Lattice-Based Proxy Re-Encryption: Reconstruction Attacks and HRA-Secure Constructions

Haotian Yin[1], Jie Zhang[1], Yuji Dong[2], Dominik Wojtczak[3] and Eng Gee Lim[1]

[1] Xi'an Jiaotong-Liverpool University, School of Advanced Technology, Suzhou, China
[2] Xi'an Jiaotong-Liverpool University, School of Internet of Things, Suzhou, China
[3] University of Liverpool, Department of Computer Science, Liverpool, United Kingdom

Abstract. Proxy re-encryption (PRE) allows a proxy to transform a ciphertext under one user's public key into a ciphertext under another user's public key without learning the message. Honest re-encryption attack (HRA) security requires message indistinguishability even when an adversary observes many honestly generated re-encryptions. Reusing a re-encryption key can create correlations that are invisible in a single response. We show how these correlations permit functional reconstruction in the schemes of Susilo et al. (ESORICS 2021) and Fan–Liu (ACNS 2019) under persistent key reuse, with additional noise and modulus conditions for Fan–Liu. Building on pads fixed for each source ciphertext, we then construct public-key, unidirectional PRE for any fixed constant number of hops under Gaussian search-LWE (with uniform secrets and discrete Gaussian errors) and PRF security. The construction first achieves HRA security when delegations between honest users follow a fixed order. A generic key-separation compiler removes this order by giving each user an independent key pair at each depth: two pairs for one hop and $t + 1$ for t hops. Users may delegate along cycles while all key dependencies remain acyclic. Security holds in a stateful HRA model with static corruption, and both the noise widths and ciphertext modulus are polynomial.

Keywords: Proxy re-encryption · HRA security · Learning with errors · Computational padding

1 Introduction

Proxy re-encryption (PRE) lets a proxy transform a ciphertext for a new recipient without learning its message [BBS98, AFGH06]. The ciphertext owner supplies a re-encryption key (rekey), and the proxy can then transform stored ciphertexts without further help from the owner. Honest re-encryption attack (HRA) security addresses a basic concern in this setting: a recipient who obtains many honestly generated transformations should learn nothing about messages that have not been delegated to it [Coh19]. The recipient knows its own secret key and may compare source ciphertexts with their transformed outputs.

Repeated transformations matter because their outputs share information. A rekey may be reused for many ciphertexts, and several re-encryptions of one ciphertext inherit the same encryption error. A proof must account for these relations across the entire transcript.

E-mail: haotian.yin23@student.xjtlu.edu.cn (Haotian Yin), jie.zhang01@xjtlu.edu.cn (Jie Zhang), yuji.dong02@xjtlu.edu.cn (Yuji Dong), d.wojtczak@liverpool.ac.uk (Dominik Wojtczak), enggee.lim@xjtlu.edu.cn (Eng Gee Lim)

This work is licensed under a "CC BY 4.0" license.



Showing that each response looks secure on its own is insufficient. In lattice-based PRE, the relations can be especially direct: honest input–output pairs may reveal a fixed linear map to transform a challenge ciphertext without the proxy.

Cohen et al. [CCG+25, Appendix B] sketch an HRA key-recovery attack against the rerandomisation approach of Davidson et al. [DDLM19]. In the binary-decomposition case they analyse, honest-to-corrupted responses from one delegator expose equations involving a fixed rekey-error vector. After recovering this vector, averaging repeated transformations of the same ciphertext removes the fresh rerandomisation noise and reveals its source error, enabling secret-key recovery. Thus adding a fresh encryption of zero after key switching does not by itself establish HRA security. Our attacks concern different schemes and reconstruct the transformation of the challenge directly, using honest input–output pairs and a sample-span bound.

One way to hide such relations is noise flooding. Cohen et al. [CCG+25] use a fine-grained approach to obtain HRA-secure PRE. For negligible statistical distance, the usual flooding argument requires noise that is superpolynomially larger than the error being hidden [MS25, Section 1.1]. Computational hiding offers another route. The owner-encrypted PRE construction (oePRE) [YZD+26] uses a private pad that stays fixed for each source ciphertext. Repeated responses therefore use the same pad, so averaging fresh response noise does not remove it. That construction uses the owner's secret key to create source ciphertexts.

Here we extend this approach to public encryption and multiple hops. Public encryption introduces error that depends jointly on the key and fresh encryption randomness. Multiple hops introduce new errors from cached rekey components and fresh encryptions of input pads. We also address a separate restriction in key-switching proofs: the order imposed on delegations between honest users. Our goal is public-key PRE that handles the complete re-encryption transcript with polynomial noise and allows users to delegate in either direction.

1.1 Our Results

1. **Reconstruction from honest transcripts.** We give two attacks in which a corrupted recipient reconstructs the action of a hidden rekey on the challenge. For Susilo et al. [SDDR21], reconstruction is exact in the persistent-key completion of their selective HRA experiment. For Fan–Liu [FL19, FL17], it follows after noise averaging in the persistent-key interpretation of their experiment, under the modulus, noise, and polynomial repetition conditions stated in Appendix A. Both attacks obtain the required information from honest nonchallenge transformations.

2. **Public-key PRE with polynomial noise.** We construct a unidirectional single-hop scheme and extend it to any fixed constant hop bound t. Security follows from Gaussian search-LWE and PRF security. The base schemes allow adaptive delegations that follow a fixed registration order among honest users. Theorems 4 and 5 prove their HRA security.

3. **A key-separation compiler that removes the user order.** Given an ordered t-hop HRA-secure scheme, we give each user $t + 1$ independent key pairs and use a different pair at each ciphertext depth. Theorem 6 preserves HRA security while allowing cycles in the user delegation graph. Each delegation uses t base rekeys; ciphertext size and noise are unchanged.

1.2 Comparison with Prior Work

HRA-secure lattice PRE constructions without trapdoors. We omit the trapdoor-based fine-grained PRE schemes of Zhou et al. [ZLHZ23, ZLH24]; their efficiency and constant-

Table 1: HRA-secure lattice PRE. Our column includes the key-separation compiler. Cohen et al. support unlimited hops with bootstrapping.

	Cohen et al. [CCG+25]	oePRE [YZD+26]	This work
Source encryption	Public key	Owner secret key	Public key
Hops	Bounded	Single	Constant
Error hiding	Statistical flooding	Computational padding	Computational padding
Proxy	Stateless	Stateful	Stateful
Assumptions	RLWE	RLWE + PRF	LWE + PRF

time sampling challenges are discussed by Cohen et al. [CCG+25, Section 1.2]. The oePRE proof requires an acyclic ordering of honest users (follows [PRSV17]), whereas our key-separation compiler permits user-level cycles while keeping internal key dependencies acyclic. All three constructions assume static corruption.

Cohen et al. [CCG+25] derive HRA security from PRE-CPA security and re-encryption simulability. Their construction builds on the CPA-secure construction of Polyakov, Rohloff, Sahu, and Vaikuntanathan [PRSV17], whose CPA security proof assumes an acyclic re-encryption graph on honest keys. Consequently, that proof does not directly establish security for the cyclic honest-key delegations permitted by their Definition 7. Cohen et al. also provide homomorphic evaluation and implementation results; our comparison is asymptotic.

Fuchsbauer et al. [FKKP19] establish adaptive HRA security using a graph-pebbling reduction whose security loss depends on the class of acyclic challenge graphs. Their LWE-based instantiation uses key switching and statistical blurring, but their reduction does not cover arbitrary cyclic key dependencies.

1.3 Technique Overview

Learning a map from its outputs. After subtracting public terms, the Susilo et al. [SDDR21] transcripts have the form $\mathbf{w}_i = \mathbf{z}_i\mathbf{T}$, where $\mathbf{z}_i$ is public and $\mathbf{T}$ is fixed by the cached rekey. Stack the observed inputs and outputs into $\mathbf{Z}$ and $\mathbf{W}$. If the challenge input satisfies $\mathbf{z}^* = \mathbf{g}^\top\mathbf{Z}$, then $\mathbf{g}^\top\mathbf{W} = \mathbf{z}^*\mathbf{T}$ gives its transformed output. Thus the adversary only needs the map's action on the challenge input[1]. Lemma 3 shows that a linear number of independent samples suffices for constant success probability, even when the input distribution does not span the whole space. In Fan–Liu [FL19], repeated responses first let the adversary average and round away fresh error. The common requirement is to preserve the relations between responses when analysing security.

Handling public encryption. Our proof must hide a source ciphertext while keeping a noisy version of its error as auxiliary information. In owner-encrypted PRE [YZD+26], the source error is sampled directly. In public encryption, it also contains products of secret-key terms and encryption randomness.

We use the LP11 encryption scheme [LP11] in the form developed by Micciancio and Suhl [MS25]. Their normal-form argument gives an LWE-like ciphertext with Gaussian error of publicly known width. We make the argument fully discrete and keep the same error jointly with its ciphertext throughout the comparison. We then use the noisy-error-leakage LWE viewpoint of [YZD+26], which permits a noisy observation of the error while

[1] A direct approach is to solve for $\mathbf{T}$ from the observed pairs $(\mathbf{z}_i, \mathbf{w}_i)$.

preserving LWE pseudorandomness. Section 4.5 combines these tools into one joint hiding lemma.

For user A, let $\mathbf{A}$ be the public matrix, $\mathbf{s}$ the secret key, and $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}_{\mathsf{pk}} \pmod q$ the public-key vector, where $\mathbf{e}_{\mathsf{pk}}$ is the public-key error. The LP11 ciphertext encrypting μ_A is

$$c_A = (\mathbf{a}_A, v_A) = \left(\mathbf{A}^\top \mathbf{r} + \mathbf{f}, \langle \mathbf{b}, \mathbf{r}\rangle + \mu_A + e'\right) \pmod q.$$

For a fixed key, $\mathbf{r}, \mathbf{f} \leftarrow D_{\mathbb{Z}^n, s_{\mathsf{err}}}$ and the scalar error e' are sampled independently; the width of e' is specified below. The ciphertext noise is

$$e_{\mathsf{ct},A} := \langle \mathbf{e}_{\mathsf{pk}}, \mathbf{r}\rangle - \langle \mathbf{f}, \mathbf{s}\rangle + e',$$

so $v_A - \langle \mathbf{a}_A, \mathbf{s}\rangle \equiv \mu_A + e_{\mathsf{ct},A} \pmod q$.

The public key contains the squared norm $\kappa_{\mathsf{pk}} = \|\mathbf{s}\|_2^2 + \|\mathbf{e}_{\mathsf{pk}}\|_2^2$. This scalar is used by the encryption algorithm: with public base width s_{err}, a sender samples the extra error e' from a discrete Gaussian of width $s_{\mathsf{err}}\sqrt{\kappa_{\mathsf{pk}}}$. This matches the scale of the secret-dependent error terms and leads to the publicly known normal-form width $\sqrt{2\kappa_{\mathsf{pk}}}\, s_{\mathsf{err}}$. Publishing κ_{pk} therefore lets anyone choose the required noise width without knowing either hidden vector.

Keeping the same pad across repeated uses. To address the reconstruction attacks above, we associate each source ciphertext with a private pad that remains fixed across repeated re-encryptions. In the single-hop construction, each source encryption carries an independent public random tag. The proxy derives the pad $p_{\mathsf{ct},A}$ for c_A from this tag using a PRF and reuses it for all recipients. Each re-encryption adds a fresh encryption of this same pad, so the source error enters every response through the fixed noisy value $\ell_{\mathsf{ct},A} = e_{\mathsf{ct},A} + p_{\mathsf{ct},A}$. Repeated queries therefore do not produce independent pad samples that can be averaged away.

This noisy error is also enough to simulate a permitted response to a corrupted recipient B. Its LP11 ciphertext has the exact rewrite

$$c_B = (\mathbf{0}, \mu_A + \ell_{\mathsf{ct},A}) - \sum_j d_{A,j} K_j^{(0)} + P^{(0)} \pmod q.$$

Here μ_A is the message, $\mathbf{d}_A$ is the public gadget decomposition, $K_j^{(0)}$ are cached encryptions of zero, and $P^{(0)}$ is a fresh encryption of zero. The rewrite uses the same encryption coins as the real response. It lets the proof answer these queries from this noisy error, without the source secret key.

Choosing the pad for the next hop. A derived (re-encrypted) ciphertext contains errors from its input, the cached rekey components, and a fresh encryption of the input pad. We assign a private pad to each LP11 encryption instance: every source encryption, every cached rekey component, and every fresh encryption of an input pad. For each such instance, write $\ell_i = e_i + p_i$.

When a ciphertext moves from A to B, choose its next pad using the pads of the rekey components and the fresh encryption of the input pad:

$$p_{\mathsf{ct},B} = -\sum_j d_{A,j} p_{\mathsf{rk},j} + p_{\mathsf{fr}},$$
$$\ell_{\mathsf{ct},B} = \ell_{\mathsf{ct},A} - \sum_j d_{A,j} \ell_{\mathsf{rk},j} + \ell_{\mathsf{fr}}.$$

The second line follows from the ciphertext noise after re-encryption and the pad choice in the first line. It is the reason the proof extends across hops: each new noisy error is

computed from noisy errors already covered by the joint lemma. We apply that lemma once per honest key to all LP11 encryption instances under it. The recurrence preserves their reuse across repeated transformations and branching paths. Public encryption tags and the immediate input decomposition let the proxy recover the next pad from the same private PRF seed.

Separating the keys at different depths. The ordered restriction comes from rekeys that encrypt functions of sk_A under pk_B. If the adversary also queries rekey $B \to A$, the encryptions depend on each other's secret keys. Polyakov et al. [PRSV17] avoid this key-dependent-message problem by requiring an acyclic graph and processing keys from sinks to sources. Our base proof follows a fixed order for the same reason.

The compiler separates users from the keys used at each depth. Write a ciphertext as (h, ct), where h counts completed hops. Give user A independent pairs $(\mathsf{pk}_{A,h}, \mathsf{sk}_{A,h})$ for $h = 0, \ldots, t$, and allow base rekeys only from depth h to depth $h+1$. A user cycle $A \to B \to A$ now uses $A^{(0)} \to B^{(1)} \to A^{(2)}$, so no key dependency returns to an earlier key. For one hop, all edges go from source keys to terminal keys, giving a bipartite graph. This two-key separation also appears in the collusion-safe PRE compiler [YZL+25]; here it removes the HRA order restriction and extends to t hops.

The corresponding ciphertext records form a forest: every successful transformation creates a fresh child of one existing record, and every path has length at most t. Figure 6 distinguishes this forest from the acyclic key graph and the possibly cyclic user graph. Viewing each key pair as a separate base user gives the security reduction.

Polynomial parameters. Public-key and ciphertext hiding holds even given the noisy errors. The pad widths therefore remain polynomial, and correctness follows by bounding the error increase at each hop. For a fixed hop bound, we take $n = \lambda$, pad width $\Theta(\sqrt{n})$, encryption width $\Theta(\sqrt{n}\log^2 n)$, and prime modulus $q = \Theta(n^3)$. Except with negligible probability, every ciphertext has noise $O(n^{5/2}\log^4 n) = o(q)$.

1.4 Organisation

Section 2 defines the model, and Section 3 gives the attacks. Section 4 develops the encryption and LWE tools used in the single-hop and multi-hop constructions of Sections 5 and 6. Section 7 gives the key-separation compiler. The appendices contain the longer attack, reduction, multi-hop security, and compiler proofs.

2 Preliminaries and Security Model

We fix the Gaussian conventions, stateful PRE syntax, and HRA experiment used throughout the paper.

H2H and H2C denote honest-to-honest and honest-to-corrupted directions, respectively.

2.1 Notation and Discrete Gaussians

The security parameter is λ. Write $[r] = \{1, \ldots, r\}$, $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$, and $\perp$ for rejection. All query and user bounds are polynomials. The symbols $\approx_s$ and $\approx_c$ denote negligible statistical distance and computational indistinguishability. Vectors are columns.

We use the standard Gaussian notation of Micciancio and Regev [MR07]. For $s > 0$, discrete Gaussian $D_{\mathbb{Z}^d,s,\mathbf{c}}$ assigns mass proportional to $\rho_{s,\mathbf{c}}(\mathbf{x}) = e^{-\pi\|\mathbf{x}-\mathbf{c}\|_2^2/s^2}$ to $\mathbf{x} \in \mathbb{Z}^d$. A zero centre is omitted, i.e., write $D_{\mathbb{Z}^d,s,\mathbf{0}} = D_{\mathbb{Z}^d,s}$ and $\rho_{s,\mathbf{0}} = \rho_s$. We use the following standard subgaussian bound for discrete Gaussians [MP11, Lemma 2.8].

Lemma 1 (Discrete Gaussian subgaussianity)**.** *Let* $\mathbf{x} \leftarrow D_{\mathbb{Z}^d,s}$ *with* $s > 0$*. For every fixed* $\mathbf{u} \in \mathbb{R}^d$ *and* $t \in \mathbb{R}$*,*

$$\mathbb{E}[e^{2\pi t\langle \mathbf{u},\mathbf{x}\rangle}] \le e^{\pi s^2\|\mathbf{u}\|^2 t^2}.$$

Consequently, for $\mathbf{u} \neq 0$ *and* $B > 0$*,*

$$\Pr[|\langle \mathbf{u},\mathbf{x}\rangle| \ge B] \le 2e^{-\pi B^2/(s^2\|\mathbf{u}\|^2)}.$$

For a countable set $S \subseteq \mathbb{R}^d$, write $\rho_s(S) = \sum_{\mathbf{x}\in S}\rho_s(\mathbf{x})$. Following [MR07, Definition 3.1], the smoothing parameter of a full-rank lattice $\Lambda \subseteq \mathbb{R}^d$ with dual Λ^* is $\eta_\epsilon(\Lambda) = \inf\{s > 0 : \rho_{1/s}(\Lambda^* \setminus \{\mathbf{0}\}) \le \epsilon\}$ for $\epsilon > 0$. We use the following bound from [MR07, Lemma 3.3].

Lemma 2 (Smoothing bound)**.** *For every* $d \ge 1$ *and* $\epsilon > 0$*,*

$$\eta_\epsilon(\mathbb{Z}^d) \le \sqrt{\frac{\ln(2d(1+1/\epsilon))}{\pi}}.$$

We use the following shifted discrete-Gaussian convolution bound, a specialisation of GMPW20 [GMPW20, Theorem 4.5 of the full version].

Theorem 1 (Shifted discrete-Gaussian convolution)**.** *Let* $d = d(\lambda) \ge 1$ *be polynomially bounded, let* $0 < \epsilon = \epsilon(\lambda) < 1$ *be negligible, and let* $s_1, s_2 > 0$ *satisfy*

$$s_2 \ge \eta_\epsilon(\mathbb{Z}^d), \qquad (s_1^{-2} + s_2^{-2})^{-1/2} \ge \eta_\epsilon(\mathbb{Z}^d).$$

For any fixed centre $\mathbf{c} \in \mathbb{R}^d$*, independently sample*

$$\mathbf{x} \leftarrow D_{\mathbb{Z}^d,s_1}, \qquad \mathbf{y} \leftarrow D_{\mathbb{Z}^d,s_2,\mathbf{c}}.$$

Then

$$\mathbf{x} + \mathbf{y} \approx_s D_{\mathbb{Z}^d,\sqrt{s_1^2+s_2^2},\mathbf{c}}.$$

The specialisation uses the cosets $\mathbb{Z}^d$ and $\mathbb{Z}^d - \mathbf{c}$, followed by translation by $\mathbf{c}$.

Learning with errors. For dimensions n, m, modulus $q \ge 2$, and width $s_{\mathsf{pk}} > 0$, Gaussian search-LWE asks to recover $\mathbf{t}$ from $(\mathbf{A}, \mathbf{A}\mathbf{t} + \mathbf{e} \bmod q)$, where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m\times n}$, $\mathbf{t} \xleftarrow{\$} \mathbb{Z}_q^n$, and $\mathbf{e} \leftarrow D_{\mathbb{Z}^m,s_{\mathsf{pk}}}$ are independent. Our base LWE assumption is that no PPT algorithm solves this problem with non-negligible probability for the stated parameters and polynomially many samples.

Small-secret Gaussian decisional LWE asks to distinguish $(\mathbf{A}, \mathbf{A}\mathbf{s}+\mathbf{e} \bmod q)$ from $(\mathbf{A}, \mathbf{u})$, where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m\times n}$, $\mathbf{s} \leftarrow D_{\mathbb{Z}^n,s_{\mathsf{pk}}}$, $\mathbf{e} \leftarrow D_{\mathbb{Z}^m,s_{\mathsf{pk}}}$, and $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m$ are mutually independent.

2.2 Stateful PRE syntax and correctness

Based on the unidirectional PRE syntax of Ateniese et al. [AFGH06, Definition 2.2], we define a stateful PRE scheme with a persistent proxy state.

Definition 1 (Stateful PRE)**.** A ciphertext is written as (h, ct), where h is its depth. A stateful PRE scheme Π has the following PPT algorithms:

- $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$ generates public parameters.
- $\mathsf{ProxyInit}(\mathsf{pp}) \to \mathsf{st_P}$ initialises the private proxy state.
- $\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{pk}_A, \mathsf{sk}_A)$ generates a public-secret key pair.

- $\mathsf{Enc}(\mathsf{pp}, \mathsf{pk}_A, m) \to (0, \mathsf{ct}_A)$ encrypts a message $m \in \mathsf{M}$ under pk_A.
- $\mathsf{ReKeyGen}(\mathsf{pp}, \mathsf{sk}_A, \mathsf{pk}_B) \to \mathsf{rk}_{A\to B}$ generates a rekey from user A to user B.
- $\mathsf{ReEnc}(\mathsf{pp}, \mathsf{rk}_{A\to B}, (h, \mathsf{ct}_A), \mathsf{pk}_B; \mathsf{st}_\mathsf{P}) \to (h+1, \mathsf{ct}_B)$ or $\perp$ transforms a ciphertext under pk_A into one decryptable under sk_B, using the proxy state.
- $\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}_A, (h, \mathsf{ct})) \to m$ or $\perp$ decrypts a ciphertext or rejects.

The public parameters pp are omitted when clear from context.

Definition 2 (T-hop PRE)**.** A stateful PRE scheme is T-hop if its public parameters specify a bound $T \geq 1$, ciphertext depths lie in $\{0, \ldots, T\}$, and re-encryption rejects inputs at depth T. Invalid depth tags are rejected. A hop is one successful re-encryption, so depth counts the hops already performed. Depth-0 ciphertexts are sources, positive-depth ciphertexts are derived, and depth-T ciphertexts are terminal. A single-hop scheme has $T = 1$.

We write the depth tag explicitly to make the algorithms and proofs easy to follow. A proxy may instead keep it with each immutable ciphertext record and supply it to the algorithms when needed, rather than include it in the ciphertext encoding. In the HRA experiment below, depth is already determined by the query history, so displaying it gives no additional information.

Definition 3 (T-hop correctness)**.** Let Π be a stateful T-hop PRE scheme. For any $0 \leq h \leq T$, message $m \in \mathsf{M}$, and sequence of honestly generated users $u_0, \ldots, u_h$, generate $(0, \mathsf{ct}_0) \leftarrow \mathsf{Enc}(\mathsf{pk}_{u_0}, m)$ and successively compute $(i, \mathsf{ct}_i) \leftarrow \mathsf{ReEnc}(\mathsf{rk}_{u_{i-1}\to u_i}, (i-1, \mathsf{ct}_{i-1}), \mathsf{pk}_{u_i}; \mathsf{st}_\mathsf{P})$ for $i \in [h]$. Each rekey is honestly generated. Correctness requires that, with probability at least $1 - \mathsf{negl}(\lambda)$, all calls succeed and $\mathsf{Dec}(\mathsf{sk}_{u_i}, (i, \mathsf{ct}_i)) = m$ for every $0 \leq i \leq h$.

2.3 HRA Security

We use Cohen's honest re-encryption attack (HRA) game [Coh19, Definitions 5–6], written with explicit ciphertext records and adapted to stateful T-hop PRE.

Let $\mathsf{Users} = \mathsf{Hon} \cup \mathsf{Cor}$. Write $\mathsf{H2C}(i, j)$ for $i \in \mathsf{Hon}$ and $j \in \mathsf{Cor}$, and $\mathsf{BadEdge}(i, j)$ for $i = j$ or i, j not both in Users. Ciphertext records have globally unique identifiers k; $\mathsf{Ct}[(i, k)] = (h, \mathsf{ct})$ stores the complete ciphertext owned by user i, and $\mathsf{Msg}[(i, k)]$ its underlying message. The set Deriv contains the challenge record and all its oracle-generated descendants. Queries select immutable records by identifier. The experiment is given in Fig. 1.

Definition 4 (Stateful T-hop HRA security)**.** A stateful T-hop PRE scheme Π is HRA-secure if, for every PPT adversary $\mathcal{A}$,

$$\mathsf{Adv}^{\mathsf{hra},T}_{\Pi}(1^\lambda, \mathcal{A}) := \left| \Pr\left[\mathsf{Exp}^{\mathsf{hra},T}_{\Pi}(1^\lambda, \mathcal{A}) = 1\right] - \frac{1}{2} \right|$$

is negligible in λ. We omit T when the scheme's hop bound is clear, and call the case $T = 1$ single-hop HRA security.

3 Honest-Transcript Reconstruction Attack

This section presents the reconstruction attack based on honest re-encryption transcripts. Susilo et al. [SDDR21] expose exact linear transcripts. Fan–Liu [FL19] add fresh Gaussian noise, which repeated queries can average away under the stated conditions. We keep the original schemes' ciphertext notation.

$\underline{\mathsf{Exp}^{\mathsf{hra},T}_{\Pi}(1^\lambda,\mathcal{A})}$
$\beta \xleftarrow{\$} \{0,1\}$
$\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$; $\mathsf{st}_\mathsf{P} \leftarrow \mathsf{ProxyInit}(\mathsf{pp})$; $n_\mathsf{k}, n_\mathsf{c} \leftarrow 0$
$\mathsf{Hon}, \mathsf{Cor}, \mathsf{Ct}, \mathsf{Msg}, \mathsf{Deriv}, \mathsf{ReKeys} \leftarrow \emptyset$; $k^\star \leftarrow \bot$
$\mathsf{state} \leftarrow \mathcal{A}_1^{\mathsf{honkeygen},\mathsf{corkeygen}}(\mathsf{pp})$
$\beta' \leftarrow \mathcal{A}_2^{\mathsf{encrypt},\mathsf{challenge},\mathsf{rekey},\mathsf{reenc}}(\mathsf{pp},\mathsf{state})$; **return** $\mathbf{1}[\beta'=\beta]$

$\underline{\mathsf{honkeygen}}$
$n_\mathsf{k} \leftarrow n_\mathsf{k}+1$
$(\mathsf{pk}_{n_\mathsf{k}}, \mathsf{sk}_{n_\mathsf{k}}) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$
$\mathsf{Hon} \leftarrow \mathsf{Hon} \cup \{n_\mathsf{k}\}$
return $(n_\mathsf{k}, \mathsf{pk}_{n_\mathsf{k}})$

$\underline{\mathsf{corkeygen}}$
$n_\mathsf{k} \leftarrow n_\mathsf{k}+1$
$(\mathsf{pk}_{n_\mathsf{k}}, \mathsf{sk}_{n_\mathsf{k}}) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$
$\mathsf{Cor} \leftarrow \mathsf{Cor} \cup \{n_\mathsf{k}\}$
return $(n_\mathsf{k}, \mathsf{pk}_{n_\mathsf{k}}, \mathsf{sk}_{n_\mathsf{k}})$

$\underline{\mathsf{encrypt}(i,m)}$
if $i \notin \mathsf{Users} \vee m \notin \mathsf{M}$
  then return $\bot$
$(0,\mathsf{ct}) \leftarrow \mathsf{Enc}(\mathsf{pk}_i, m)$; $n_\mathsf{c} \leftarrow n_\mathsf{c}+1$
$\mathsf{Ct}[(i,n_\mathsf{c})] \leftarrow (0,\mathsf{ct})$
$\mathsf{Msg}[(i,n_\mathsf{c})] \leftarrow m$
return $(n_\mathsf{c}, (0,\mathsf{ct}))$

$\underline{\mathsf{challenge}(i,m_0,m_1)}$
if $i \notin \mathsf{Hon} \vee k^\star \neq \bot$
  then return $\bot$
if $m_0 \notin \mathsf{M} \vee m_1 \notin \mathsf{M}$
  then return $\bot$
$(0,\mathsf{ct}^\star) \leftarrow \mathsf{Enc}(\mathsf{pk}_i, m_\beta)$; $n_\mathsf{c} \leftarrow n_\mathsf{c}+1$
$k^\star \leftarrow n_\mathsf{c}$; $\mathsf{Ct}[(i,k^\star)] \leftarrow (0,\mathsf{ct}^\star)$
$\mathsf{Msg}[(i,k^\star)] \leftarrow m_\beta$
$\mathsf{Deriv} \leftarrow \mathsf{Deriv} \cup \{k^\star\}$
return $(k^\star, (0,\mathsf{ct}^\star))$

$\underline{\mathsf{GetRK}(i,j)}$
if $\mathsf{ReKeys}[(i,j)]$ undefined then
  $\mathsf{ReKeys}[(i,j)] \leftarrow \mathsf{ReKeyGen}(\mathsf{sk}_i, \mathsf{pk}_j)$
return $\mathsf{ReKeys}[(i,j)]$

$\underline{\mathsf{rekey}(i,j)}$
if $\mathsf{BadEdge}(i,j) \vee \mathsf{H2C}(i,j)$
  then return $\bot$
return $\mathsf{GetRK}(i,j)$

$\underline{\mathsf{reenc}(i,j,k)}$
if $\mathsf{BadEdge}(i,j) \vee \mathsf{Ct}[(i,k)]$ undefined **then return** $\bot$
$(h,\mathsf{ct}) \leftarrow \mathsf{Ct}[(i,k)]$
if $h \geq T \vee (k \in \mathsf{Deriv} \wedge j \in \mathsf{Cor})$ **then return** $\bot$
$\mathsf{out} \leftarrow \mathsf{ReEnc}(\mathsf{GetRK}(i,j), (h,\mathsf{ct}), \mathsf{pk}_j; \mathsf{st}_\mathsf{P})$
if $\mathsf{out} = \bot$ **then return** $\bot$
$n_\mathsf{c} \leftarrow n_\mathsf{c}+1$; $\mathsf{Ct}[(j,n_\mathsf{c})] \leftarrow \mathsf{out}$
$\mathsf{Msg}[(j,n_\mathsf{c})] \leftarrow \mathsf{Msg}[(i,k)]$
if $k \in \mathsf{Deriv}$ **then** $\mathsf{Deriv} \leftarrow \mathsf{Deriv} \cup \{n_\mathsf{c}\}$
return $(n_\mathsf{c}, \mathsf{out})$

Figure 1: HRA experiment for stateful T-hop PRE. GetRK is an internal helper.

3.1 Direct Reconstruction against Susilo et al.

We analyse Susilo et al.'s attribute-based PRE scheme abPRE [SDDR21].

3.1.1 Scheme overview and exposed linear structure

In abPRE, a ciphertext is associated with an attribute vector $\mathbf{x} \in \mathbb{Z}_q^{d_{\mathsf{attr}}}$. A secret key for a policy f decrypts ciphertexts satisfying $f(\mathbf{x}) = 0$, while a rekey from f to g transforms such a ciphertext into one decryptable under a policy key for g.

We describe only the algebraic structure relevant to the attack. Let m_{AB} denote the ciphertext-block dimension and define $d_{\mathsf{rec}} := 2m_{\mathsf{AB}}\lceil \log q \rceil$. Assume that q is prime, so that $\mathbb{Z}_q = \mathbb{F}_q$; all arithmetic below is over this field.

An input ciphertext has the form $\mathsf{ct} = (\mathbf{c}_{\mathrm{in}}, \mathbf{c}_1, \ldots, \mathbf{c}_{d_{\mathsf{attr}}}, \mathbf{c}_{\mathrm{out}})$. The proxy evaluates f to obtain $\mathbf{c}_f$ and computes the publicly derivable row vector $\mathbf{z} := \mathsf{BD}((\mathbf{c}'_f)^\top) \in \{0,1\}^{1\times d_{\mathsf{rec}}}$, where $\mathbf{c}'_f := [\mathbf{c}_{\mathrm{in}} \mid \mathbf{c}_f]$.

The rekey has the block form

$$\mathsf{rk}_{f\to g} = \begin{bmatrix} \mathbf{K}_{\mathsf{body}} & \mathbf{K}_{\mathsf{out}} \\ \mathbf{0} & \mathbf{I}_{m_{\mathsf{AB}}} \end{bmatrix},$$

where $\mathbf{K}_{\mathsf{body}} \in \mathbb{F}_q^{d_{\mathsf{rec}} \times (d_{\mathsf{attr}}+1)m_{\mathsf{AB}}}$ and $\mathbf{K}_{\mathsf{out}} \in \mathbb{F}_q^{d_{\mathsf{rec}} \times m_{\mathsf{AB}}}$. Writing the output as $\mathsf{ct}' = (\mathbf{c}'_{\mathsf{body}}, \mathbf{c}'_{\mathsf{out}})$, re-encryption gives $(\mathbf{c}'_{\mathsf{body}})^\top = \mathbf{z}\mathbf{K}_{\mathsf{body}}$ and $(\mathbf{c}'_{\mathsf{out}})^\top = \mathbf{z}\mathbf{K}_{\mathsf{out}} + \mathbf{c}_{\mathrm{out}}^\top$.

Define $\mathbf{T} := [\mathbf{K}_{\mathsf{body}} \mid \mathbf{K}_{\mathsf{out}}]$ and the publicly computable adjusted response $\mathbf{w}(\mathsf{ct}, \mathsf{ct}') := [(\mathbf{c}'_{\mathsf{body}})^\top \mid (\mathbf{c}'_{\mathsf{out}})^\top - \mathbf{c}_{\mathrm{out}}^\top]$. Every honest re-encryption pair then satisfies

$$\mathbf{w}(\mathsf{ct}, \mathsf{ct}') = \mathbf{z}\mathbf{T}. \tag{1}$$

Every query for the delegation (f, g) uses the same key. Thus polynomially many honest replies reveal evaluations of one fixed hidden linear map $\mathbf{T}$.

3.1.2 Reconstruction attack

Choose $f, g, \mathbf{x}^*$ with $f(\mathbf{x}^*) = 0$ and $g(\mathbf{x}^*) \neq 0$, where g accepts some attribute. The selective HRA game of Susilo et al. [SDDR21, Section 3] permits obtaining sk_g and honest encryptions under $\mathbf{x}^*$ with their pre-challenge re-encryptions, but rejects the rekey query $f \to g$. The attack never requests this rekey; it reconstructs only its action on the challenge.

1. **Setup.** Obtain sk_g and choose distinct equal-length messages m_0, m_1. The rekey is generated once, fixing its output attribute $\mathbf{y}$ with $g(\mathbf{y}) = 0$.

2. **Collect transcripts.** Before the challenge, obtain N_{samp} honest encryptions ct_i of zero under $\mathbf{x}^*$ and their re-encryptions ct'_i from f to g using the same rekey. Stack the public vectors $\mathbf{z}_i$ and $\mathbf{w}(\mathsf{ct}_i, \mathsf{ct}'_i)$ as the rows of $\mathbf{Z}$ and $\mathbf{W}$. By Equation (1),

$$\mathbf{W} = \mathbf{Z}\mathbf{T}. \tag{2}$$

3. **Reconstruct and decrypt.** Given ct^*, compute $\mathbf{z}^*$ and solve $\mathbf{z}^* = \mathbf{g}_{\mathsf{rec}}^\top \mathbf{Z}$ for $\mathbf{g}_{\mathsf{rec}} \in \mathbb{F}_q^{N_{\mathsf{samp}}}$, guessing uniformly if no solution exists. Otherwise, compute $\mathbf{w}^* := \mathbf{g}_{\mathsf{rec}}^\top \mathbf{W}$ and add back $\mathbf{c}_{\mathrm{out}}^{*\top}$ to its output block to assemble the re-encrypted ciphertext. Decrypt with sk_g and output the bit corresponding to the recovered message.

Whenever $\mathbf{z}^* \in \mathrm{RowSpan}_{\mathbb{F}_q}(\mathbf{Z})$, Equation (2) gives

$$\mathbf{w}^* = \mathbf{g}_{\mathsf{rec}}^\top \mathbf{W} = \mathbf{g}_{\mathsf{rec}}^\top \mathbf{Z}\mathbf{T} = \mathbf{z}^*\mathbf{T}.$$

Thus adding back $\mathbf{c}_{\mathrm{out}}^{*\top}$ yields the exact honest re-encryption of the challenge ciphertext.

3.1.3 Success condition and probability

The plaintext affects only $\mathbf{c}_{\mathrm{out}}$, while $\mathbf{z} = \mathsf{BD}((\mathbf{c}'_f)^\top)$ depends on $\mathbf{c}_{\mathrm{in}}$ and $\mathbf{c}_f$. Fix the public parameters, user keys, and challenge attribute $\mathbf{x}^*$. Fresh encryption randomness then makes $\mathbf{z}_1, \ldots, \mathbf{z}_{N_{\mathsf{samp}}}, \mathbf{z}^*$ independent samples from the same message-independent distribution. Define

$$\mathsf{Good}_{\mathsf{span}} := \left\{ \mathbf{z}^* \in \mathrm{RowSpan}_{\mathbb{F}_q}(\mathbf{Z}) \right\}.$$

The attack reconstructs the honest re-encryption of the challenge ciphertext whenever $\mathsf{Good}_{\mathsf{span}}$ occurs.

Lemma 3 (Sample-span bound)**.** *Let $\mathbb{F}$ be any finite field, let D be any distribution over $\mathbb{F}^{d_{\mathsf{rec}}}$, and let $\mathbf{z}_1, \ldots, \mathbf{z}_{N_{\mathsf{samp}}}, \mathbf{z}^*$ be independent samples from D. Let $r := \dim(\mathrm{span}_{\mathbb{F}}(\mathrm{supp}(D)))$. Then*

$$\Pr\left[\mathbf{z}^* \in \mathrm{span}_{\mathbb{F}}\{\mathbf{z}_1, \ldots, \mathbf{z}_{N_{\mathsf{samp}}}\}\right] \geq 1 - \frac{r}{N_{\mathsf{samp}} + 1} \geq 1 - \frac{d_{\mathsf{rec}}}{N_{\mathsf{samp}} + 1}.$$

Proof. Set $\mathbf{z}_{N_{\mathsf{samp}}+1} := \mathbf{z}^*$. For each $j \in [N_{\mathsf{samp}} + 1]$, let E_j be the event that $\mathbf{z}_j$ does not lie in the span of the remaining vectors. For every fixed realisation, the vectors whose indices satisfy E_j are linearly independent: any linear dependence among them would express one such vector as a linear combination of vectors among the remaining positions, contradicting E_j. Hence at most r positions satisfy E_j, and therefore $\sum_{j=1}^{N_{\mathsf{samp}}+1} \mathbf{1}[E_j] \le r$. Taking expectations and using exchangeability gives $(N_{\mathsf{samp}} + 1)\Pr[E_{N_{\mathsf{samp}}+1}] \le r$. Since $E_{N_{\mathsf{samp}}+1}$ is precisely the event that $\mathbf{z}^* \notin \mathrm{span}_{\mathbb{F}}\{\mathbf{z}_1, \ldots, \mathbf{z}_{N_{\mathsf{samp}}}\}$, taking complements proves the first inequality. The second follows from $r \le d_{\mathsf{rec}}$. □

Applying Lemma 3 over $\mathbb{F}_q$ with $N_{\mathsf{samp}} = 2d_{\mathsf{rec}}$ gives

$$\Pr[\mathsf{Good}_{\mathsf{span}}] \ge 1 - \frac{d_{\mathsf{rec}}}{2d_{\mathsf{rec}} + 1} = \frac{d_{\mathsf{rec}} + 1}{2d_{\mathsf{rec}} + 1} > \frac{1}{2}. \tag{3}$$

Theorem 2 (Persistent-key HRA insecurity of abPRE)**.** *Assume that q is prime, $d_{\mathsf{rec}} = \mathsf{poly}(\lambda)$, and the correctness error is negligible. Assume that the policy family contains $f, g, \mathbf{x}^*, \mathbf{y}$ as above.* abPRE *is insecure in the HRA model.*

Proof. The reconstruction in Section 3.1.2 gives exact challenge re-encryption whenever $\mathsf{Good}_{\mathsf{span}}$ occurs. By Equation (3), this event occurs with probability greater than $1/2$. Couple the reconstructed output with the real re-encryption of the challenge using the same rekey. Their outputs agree on $\mathsf{Good}_{\mathsf{span}}$; the probability that this real output decrypts incorrectly is at most the ordinary correctness error $\varepsilon_{\mathsf{cor}}$, without conditioning the correctness guarantee on the span event. On span failure the adversary guesses uniformly. Thus its guessing advantage $|\Pr[b' = b] - 1/2|$ is at least $\Pr[\mathsf{Good}_{\mathsf{span}}]/2 - \varepsilon_{\mathsf{cor}} > 1/4 - \mathsf{negl}(\lambda)$. The attack uses $2d_{\mathsf{rec}}$ honest encryption and re-encryption queries and is polynomial time. □

3.1.4 Flaw in the previous HRA proof

Because the same rekey is reused for each delegation, the real transcript satisfies $\mathbf{W} = \mathbf{ZT}$ for one fixed hidden linear map $\mathbf{T}$. The previous proof [SDDR21, Theorem 4] instead samples independent dummy matrices for different responses. Matching individual output distributions does not establish indistinguishability of the joint transcripts: the simulation must also preserve the common linear relation across queries. Our reconstruction attack exploits this missing consistency.

Implications for related constructions. Related HRA-secure lattice PRE variants include conditional [ZWW23, YWW+25b, YWW+25a], threshold [ZWX+24b], time-based [ZWX+24a], and puncturable schemes [ZJL+25, ZCL26]. The CCA1 construction of [Kir14] also uses a fixed linear re-encryption layer. Whether these constructions expose exploitable correlations under rekey reuse remains to be analysed.

Zhao et al. [ZCA+25, Section 6] adopt the HRA security model and proof framework of Susilo et al. in their attribute-based homomorphic PRE extension. Their re-encryption algorithm applies bootstrapping after the linear transformation, so our reconstruction attack does not directly extend to the final outputs. Their security proof must still account for correlations across repeated re-encryptions under the same key.

3.2 Noise-Averaged Reconstruction against Fan–Liu

Fan–Liu's single-hop tag-based scheme tbPRE [FL19, FL17] transforms a source ciphertext $(u, \mathbf{b}, 1)$ into $(u, \mathbf{b}', 2)$. Under the noise-bearing interpretation of its expanded formula,

$$(\mathbf{b}')^\top = \mathbf{b}^\top \mathsf{rk}_{A\to B} + \mathbf{e}_{\mathsf{re}}^\top \pmod{2q}, \qquad \mathbf{e}_{\mathsf{re}} \leftarrow D_{\mathbb{Z}^{3nk}, s_0},$$

where u is the public tag, $\mathbf{b} \in \mathbb{Z}_{2q}^{3nk}$, n is the lattice dimension, $k = O(\log q)$, and s_0 is the public Gaussian width. The rekey is fixed for the delegation, while fresh error is sampled on each call.

In their tag-based CCA experiment, the adversary corrupts recipient B and repeatedly requests re-encryption of independent honest encryptions under honest source key pk_A. All queries precede the challenge; neither the hidden rekey nor a challenge re-encryption is requested. Keeping the last two output blocks and subtracting the third input block from the third output block gives $(\mathbf{w}_i^{(\nu)})^\top = \mathbf{z}_i^\top \mathbf{T}_{A\to B} + (\mathbf{e}_i^{(\nu)})^\top \bmod 2q$, where $\mathbf{z}_i$ contains the first two input blocks and $\mathbf{T}_{A\to B}$ is fixed.

The attack aligns repeated responses using centred modular differences, then averages and rounds to recover the exact evaluations $\mathbf{z}_i^\top \mathbf{T}_{A\to B}$. It applies the functional reconstruction of Section 3.1 over $\mathbb{F}_2$ and $\mathbb{F}_q$, combining the solutions by CRT to compute the map's action on the challenge. Under the modulus and noise conditions of Theorem 7, this gives a polynomial-time attack with advantage greater than $1/4 - \mathsf{negl}(\lambda)$. Appendix A gives the block decomposition, parameter conditions, and proof.

4 LP11-Style Encryption and Security Tools

This section develops the encryption and distributional tools used by the PRE construction in Section 5.

4.1 Discrete LP11-Style Public-Key Encryption

Parameters and assumption. Throughout the construction and its proofs, set $n = \lambda$ and choose the following parameters with fixed implicit constants, taking that of s_{err} sufficiently large relative to that of s_{pk}:

$$\begin{aligned} s_{\mathsf{pk}} = \Theta(\sqrt{n}), \qquad s_{\mathsf{err}} = \Theta(\sqrt{n}(\log n)^2), \qquad s_{\mathsf{pad}} = 2s_{\mathsf{pk}}, \\ q \in [n^3, 2n^3] \text{ prime}, \qquad B_{\mathsf{pad}} = \lceil 2s_{\mathsf{pk}} \log n \rceil. \end{aligned} \tag{4}$$

The computational comparisons below assume Gaussian search-LWE with dimension n, modulus q, error width s_{pk}, and polynomially many samples. We use the negligible tolerance $\epsilon(n) := \exp(-(\log n)^2)$, for which Lemma 2 gives $\eta_\epsilon(\mathbb{Z}^m) = O(\log n)$ for every polynomial dimension m. All parameter inequalities below hold for sufficiently large n.

We use the Lindner–Peikert (LP11) encryption scheme [LP11] as modified by Micciancio and Suhl (MS25) [MS25, Section 5.1], replacing the continuous scalar encryption error with a discrete Gaussian.

We use a square public-key matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times n}$, where n is the LWE secret dimension. Figure 2 specifies $\mathsf{KeyGen}_{\mathsf{lp}}$ and $\mathsf{Enc}_{\mathsf{lp}}$ for payloads $\mu \in \mathbb{Z}_q$, with ciphertext space $\mathbb{Z}_q^n \times \mathbb{Z}_q$.[2] Throughout the construction and its analysis, we abort whenever a sampled key has $\kappa_{\mathsf{pk}} = 0$; this event has negligible probability.

Ciphertext Phase and Noise. Define the hidden secret vector

$$\mathbf{w} := (\mathbf{e}_{\mathsf{pk}}, -\mathbf{s}) \in \mathbb{Z}^{2n}, \qquad \|\mathbf{w}\|_2^2 = \kappa_{\mathsf{pk}}.$$

For $\mathsf{ct} = (\mathbf{a}, v)$, define its *ciphertext phase* by $\phi_{\mathbf{s}}(\mathsf{ct}) := v - \langle \mathbf{a}, \mathbf{s}\rangle \pmod q$. For $\mathbf{x} := (\mathbf{r}, \mathbf{f}) \leftarrow D_{\mathbb{Z}^{2n}, s_{\mathsf{err}}}$, define its *ciphertext noise* $e_{\mathsf{ct}} \in \mathbb{Z}$ by

$$\begin{aligned} e_{\mathsf{ct}} &:= \langle \mathbf{e}_{\mathsf{pk}}, \mathbf{r}\rangle - \langle \mathbf{f}, \mathbf{s}\rangle + e' = \langle \mathbf{w}, \mathbf{x}\rangle + e', \\ \phi_{\mathbf{s}}(\mathsf{ct}) &\equiv \mu + e_{\mathsf{ct}} \pmod q. \end{aligned} \tag{5}$$

[2] We use an external message space M with $|\mathsf{M}| = O(1)$, encoded in $\mathbb{Z}_q$ with minimum spacing $\Theta(q)$. As in MS25 [MS25, Section 2.3, footnote 4], external messages are written in their encoded form. The final decoding step in Dec is implicit.

Public parameters $n, q, s_{\mathsf{pk}}, s_{\mathsf{err}}$ are implicit.

$\underline{\mathsf{KeyGen}_{\mathsf{lp}}}$

$1 : \mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times n}$
$2 : \mathbf{s}, \mathbf{e}_{\mathsf{pk}} \leftarrow D_{\mathbb{Z}^n, s_{\mathsf{pk}}}$
$3 : \mathbf{b} \leftarrow \mathbf{A}\mathbf{s} + \mathbf{e}_{\mathsf{pk}} \pmod q$
$4 : \kappa_{\mathsf{pk}} \leftarrow \|\mathbf{s}\|_2^2 + \|\mathbf{e}_{\mathsf{pk}}\|_2^2$
$5 : \mathsf{pk} \leftarrow (\mathbf{A}, \mathbf{b}, \kappa_{\mathsf{pk}})$
$6 : \mathsf{sk} \leftarrow \mathbf{s}$
$7 :$ **return** $(\mathsf{pk}, \mathsf{sk})$

$\underline{\mathsf{Enc}_{\mathsf{lp}}(\mathsf{pk}, \mu)}$

$1 : (\mathbf{A}, \mathbf{b}, \kappa_{\mathsf{pk}}) \leftarrow \mathsf{pk}$
$2 : \mathbf{r}, \mathbf{f} \leftarrow D_{\mathbb{Z}^n, s_{\mathsf{err}}}$
$3 : e' \leftarrow D_{\mathbb{Z}, s_{\mathsf{err}}\sqrt{\kappa_{\mathsf{pk}}}}$
$4 : \mathbf{a} \leftarrow \mathbf{A}^\top \mathbf{r} + \mathbf{f} \pmod q$
$5 : v \leftarrow \langle \mathbf{b}, \mathbf{r} \rangle + \mu + e' \pmod q$
$6 :$ **return** $\mathsf{ct} = (\mathbf{a}, v)$

Figure 2: Discrete LP11-style key generation and public-key encryption.

Ciphertext-noise bound. For any fixed honestly generated key with $\kappa_{\mathsf{pk}} > 0$, the two independent terms in $e_{\mathsf{ct}} = \langle \mathbf{w}, \mathbf{x} \rangle + e'$ are each zero-subgaussian with parameter $s_{\mathsf{err}}\sqrt{\kappa_{\mathsf{pk}}}$ by Lemma 1. Consequently, e_{ct} is zero-subgaussian with parameter $s_{\mathsf{ct}}(\mathsf{pk}) := \sqrt{2\kappa_{\mathsf{pk}}}\, s_{\mathsf{err}}$. Its two-sided tail bound gives, for every $B > 0$,

$$\Pr[|e_{\mathsf{ct}}| \geq B \mid \mathbf{w}] \leq 2\exp\left(-\frac{\pi B^2}{s_{\mathsf{ct}}(\mathsf{pk})^2}\right). \tag{6}$$

4.2 Public-Norm Leakage and LWE Pseudorandomness

The published key norm is correlated with the secret and error, so the Gaussian search-LWE assumption does not directly give the required joint pseudorandomness. We adapt MS25's known-norm techniques and arbitrary-error search-to-decision reduction, then use regular HNF to obtain the public-key and source blocks. The complete proof, including norm bounds, is given in Appendix B.

Lemma 4 (Full source/key block pseudorandomness)**.** *For any polynomially bounded $M \geq 0$, sample $(\mathbf{s}, \mathbf{e}_{\mathsf{pk}}, \mathbf{e}_{\mathsf{base}}) \leftarrow D_{\mathbb{Z}^{2n+M}, s_{\mathsf{pk}}}$ with block lengths n, n, M, and set $\kappa_{\mathsf{pk}} = \|\mathbf{s}\|_2^2 + \|\mathbf{e}_{\mathsf{pk}}\|_2^2$. Sample $\mathbf{A}_0 \xleftarrow{\$} \mathbb{Z}_q^{n\times n}$, $\mathbf{A}_1 \xleftarrow{\$} \mathbb{Z}_q^{M\times n}$, $\mathbf{u}_0 \xleftarrow{\$} \mathbb{Z}_q^n$, and $\mathbf{u}_1 \xleftarrow{\$} \mathbb{Z}_q^M$. Then, with arithmetic modulo q,*

$$G_{\mathsf{RR}} := (\mathbf{A}_0, \mathbf{A}_0\mathbf{s} + \mathbf{e}_{\mathsf{pk}}, \kappa_{\mathsf{pk}}, \mathbf{A}_1, \mathbf{A}_1\mathbf{s} + \mathbf{e}_{\mathsf{base}}) \approx_c G_{\mathsf{UU}} := (\mathbf{A}_0, \mathbf{u}_0, \kappa_{\mathsf{pk}}, \mathbf{A}_1, \mathbf{u}_1). \tag{7}$$

Taking $M = 0$ in Lemma 4 and discarding κ_{pk} gives n-sample small-secret Gaussian decisional LWE with secret and error width s_{pk}.

4.3 Fully Discrete LP11 Normal Form

The security proof needs a joint normal form for each ciphertext and its integer noise. We adapt MS25 [MS25, Theorem 2] to fully discrete Gaussians, keeping the same noise in the ciphertext phase and the auxiliary noise coordinate.

Lemma 5 (Fully discrete LP11 normal form with explicit noise)**.** *Generate $(\mathsf{pk}, \mathbf{s}) \leftarrow \mathsf{KeyGen}_{\mathsf{lp}}$. For any polynomial $M \geq 0$ and payloads $\mu_i \in \mathbb{Z}_q$, let $\mathsf{ct}_i \leftarrow \mathsf{Enc}_{\mathsf{lp}}(\mathsf{pk}, \mu_i)$, with ciphertext noise $e_{\mathsf{ct},i}$ defined by (5).*

Conditional on the key, sample $\mathbf{u}_i \xleftarrow{\$} \mathbb{Z}_q^n$ and $\widetilde{e}_{\mathsf{ct},i} \leftarrow D_{\mathbb{Z}, s_{\mathsf{ct}}(\mathsf{pk})}$ for each $i \in [M]$, and set $\widetilde{\mathsf{ct}}_i := (\mathbf{u}_i, \langle \mathbf{u}_i, \mathbf{s} \rangle + \mu_i + \widetilde{e}_{\mathsf{ct},i}) \bmod q$. Then

$$(\mathsf{pk}, \{(\mathsf{ct}_i, e_{\mathsf{ct},i})\}_{i=1}^M) \approx_c (\mathsf{pk}, \{(\widetilde{\mathsf{ct}}_i, \widetilde{e}_{\mathsf{ct},i})\}_{i=1}^M). \tag{8}$$

Appendix C gives the proof using discrete Gaussian projection, exact conditional sampling, and fixed-matrix shifted LWE.

4.4 Noisy-Error-Leakage LWE

For an error e and an independent Gaussian pad p, call $\ell = e + p$ its *noisy error*, or *noisy-error leakage (NEL)* when kept as proof-side auxiliary information. NEL-LWE asks whether LWE remains pseudorandom given this leakage. We recall the NEL-LWE consequence of Leaky-LWE [LSW25] from [YZD+26, Corollary 8], specialised to $R = \mathbb{Z}$.

Theorem 3 (NEL-LWE from LWE)**.** *Fix polynomially bounded dimensions n, m, modulus q, and a distribution χ_{sec}. Let $\epsilon > 0$ be negligible. Assume Gaussian decisional LWE with secret distribution χ_{sec} and error width s_{lwe}. Set $a_{\mathsf{nel}} := s_{\mathsf{lwe}}^2 + 2\eta_\epsilon(\mathbb{Z}^m)^2$ and suppose*

$$s_{\mathsf{lwe}} \geq \sqrt{2}\eta_\epsilon(\mathbb{Z}^m), \qquad s_{\mathsf{nel}}^2 > a_{\mathsf{nel}}, \qquad s_{\mathsf{pad}}^2 \geq \frac{s_{\mathsf{nel}}^2 a_{\mathsf{nel}}}{s_{\mathsf{nel}}^2 - a_{\mathsf{nel}}}. \tag{9}$$

Sample $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m\times n}$, $\mathbf{s} \leftarrow \chi_{\mathsf{sec}}$, $\mathbf{e} \leftarrow D_{\mathbb{Z}^m, s_{\mathsf{nel}}}$, $\mathbf{p} \leftarrow D_{\mathbb{Z}^m, s_{\mathsf{pad}}}$, and $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m$. Then

$$(\mathbf{A}, \mathbf{As} + \mathbf{e} \bmod q, \mathbf{e} + \mathbf{p}) \approx_c (\mathbf{A}, \mathbf{u}, \mathbf{e} + \mathbf{p}). \tag{10}$$

In our application, $\chi_{\mathsf{sec}} = D_{\mathbb{Z}^n, s_{\mathsf{pk}}}$: the LWE secret is the recipient's secret key $\mathbf{s}_B$, and the error vector consists of the ciphertext noises in the LP11 normal form, with width $s_{\mathsf{ct}}(\mathsf{pk}_B)$.

4.5 Joint Public-Key and Ciphertext Hiding

Lemma 5 puts LP11 ciphertexts and their noise in LWE form. We combine this normal form with NEL-LWE and the full-block pseudorandomness of Lemma 4 to hide the public key and ciphertexts jointly.

Lemma 6 (Joint public-key and ciphertext hiding with NEL)**.** *Generate $(\mathsf{pk}_B, \mathbf{s}_B) \leftarrow \mathsf{KeyGen}_{\mathsf{lp}}$. For any polynomial $M \geq 0$, fix payloads $\mu_i \in \mathbb{Z}_q$ independently of the encryption randomness. Set $\mathsf{ct}_i \leftarrow \mathsf{Enc}_{\mathsf{lp}}(\mathsf{pk}_B, \mu_i)$ and $\ell_i := e_{\mathsf{ct},i} + p_i$, with pads $p_i \leftarrow D_{\mathbb{Z}, s_{\mathsf{pad}}}$.*

Keeping the honestly sampled norm κ_{pk}, define $\widetilde{\mathsf{pk}}_B := (\mathbf{A}_0, \mathbf{u}_0, \kappa_{\mathsf{pk}})$, where $\mathbf{A}_0 \xleftarrow{\$} \mathbb{Z}_q^{n\times n}$ and $\mathbf{u}_0 \xleftarrow{\$} \mathbb{Z}_q^n$. Conditional on this key, sample $U_i \xleftarrow{\$} \mathbb{Z}_q^n \times \mathbb{Z}_q$ and $\widetilde{e}_i \leftarrow D_{\mathbb{Z}, s_{\mathsf{ct}}(\widetilde{\mathsf{pk}}_B)}$, and set $\widetilde{\ell}_i := \widetilde{e}_i + p_i$. Then

$$(\mathsf{pk}_B, \{(\mathsf{ct}_i, \ell_i)\}_{i=1}^M) \approx_c (\widetilde{\mathsf{pk}}_B, \{(U_i, \widetilde{\ell}_i)\}_{i=1}^M). \tag{11}$$

Appendix C.3 proves this lemma.

5 Single-Hop PRE

We build on the padding method of owner-encrypted PRE [YZD+26], using the discrete LP11 encryption layer of Section 4.1.

5.1 Ordered Single-Hop HRA Setting

We analyse the construction in the single-hop ($T = 1$) HRA experiment of Section 2.3, with an additional restriction on honest-to-honest (H2H) delegation. Honest users are assigned a fixed order $v_1 \prec \cdots \prec v_H$ according to their registration order, where $\mathrm{rank}(v_i) = i$ is determined before generating the corresponding key pair. During the query phase, the adversary may adaptively select re-encryption queries, including after the challenge, but an H2H query $\mathsf{rekey}(i, j)$ or $\mathsf{reenc}(i, j, k)$ is accepted only if $\mathrm{rank}(i) < \mathrm{rank}(j)$. Thus, every admissible H2H edge follows the predetermined order, ensuring that the honest-user delegation graph is acyclic. All other rules of Fig. 1 remain unchanged.

Definition 5 (Ordered single-hop HRA security)**.** Let $\mathsf{Exp}^{\mathsf{ord\text{-}hra}}_{\Pi}$ be the experiment above with $T = 1$. A scheme Π is ordered single-hop HRA-secure if, for every PPT adversary $\mathcal{A}$,

$$\mathsf{Adv}^{\mathsf{ord\text{-}hra}}_{\Pi}(1^\lambda, \mathcal{A}) := \left| \Pr\left[\mathsf{Exp}^{\mathsf{ord\text{-}hra}}_{\Pi}(1^\lambda, \mathcal{A}) = 1\right] - \frac{1}{2} \right|$$

is negligible.

5.2 Construction

Proxy state and pads. Let $F_{\kappa_{\mathsf{pad}}}$ be a secure PRF whose private key is held by the proxy. Its output supplies the random coins of an efficient deterministic sampler $\mathsf{Sample}_{\mathsf{pad}}$, with values in $[-B_{\mathsf{pad}}, B_{\mathsf{pad}}] \cap \mathbb{Z}$ and distribution negligibly close to $D_{\mathbb{Z},s_{\mathsf{pad}}}$ under uniform coins.

A ciphertext is (h, ct) with $\mathsf{ct} = (c, \sigma)$, where c is the underlying LP11 ciphertext. Each source encryption independently samples a public tag $\nu_{\mathsf{src}} \xleftarrow{\$} \{0,1\}^\lambda$. The header is $\sigma = \nu_{\mathsf{src}}$ at depth 0 and $\sigma = \perp$ at depth 1. The random tag identifies the source encryption instance; h records its depth. Using an injective, domain-separated encoding of the input tuple, the proxy derives the pad as

$$p_{\mathsf{ct}} := \mathsf{Sample}_{\mathsf{pad}}\left(F_{\kappa_{\mathsf{pad}}}(\mathtt{src}, \nu_{\mathsf{src}})\right). \tag{12}$$

Its noisy error is $\ell_{\mathsf{ct}} := e_{\mathsf{ct}} + p_{\mathsf{ct}}$. For a particular source $(0, \mathsf{ct}_A)$, write $p_{\mathsf{ct},A}$ and $\ell_{\mathsf{ct},A}$ for these values.

Gadget decomposition and algorithms. Re-encryption uses gadget-based key switching [BV11], encrypting gadget-weighted components of the source secret under the recipient's public key, together with a fresh encryption of the input pad. Both use the LP11-style encryption layer of Section 4.1.

Following the standard gadget construction [MP12], let $k = \lceil \log_2 q \rceil$ (binary gadget) and $\ell_{\mathsf{g}} := nk$ be the number of gadget columns. Define the gadget matrix $\mathbf{G} = \mathbf{I}_n \otimes (1, 2, \ldots, 2^{k-1}) \in \mathbb{Z}_q^{n \times \ell_{\mathsf{g}}}$. For any $\mathbf{a} \in \mathbb{Z}_q^n$, let $\mathbf{d} = \mathsf{Decomp}(\mathbf{a}) \in \{0,1\}^{\ell_{\mathsf{g}}}$ be the concatenation of the binary decompositions of its canonical representatives. Then $\mathbf{Gd} = \mathbf{a} \pmod q$, $\|\mathbf{d}\|_1 \le \ell_{\mathsf{g}}$.

5.3 Correctness and Inherited Noise

Let $e_{\mathsf{rk},j}$ and e_{fr} denote the ciphertext noise in K_j and P, respectively. Since $\phi_{\mathbf{s}_B}(K_j) \equiv \langle \mathbf{g}_j, \mathbf{s}_A \rangle + e_{\mathsf{rk},j} \pmod q$ and $\phi_{\mathbf{s}_B}(P) \equiv p_{\mathsf{ct},A} + e_{\mathsf{fr}} \pmod q$, the output satisfies

$$\begin{aligned} \phi_{\mathbf{s}_B}(c_B) &\equiv v - \sum_{j=1}^{\ell_{\mathsf{g}}} d_j\left(\langle \mathbf{g}_j, \mathbf{s}_A \rangle + e_{\mathsf{rk},j}\right) + p_{\mathsf{ct},A} + e_{\mathsf{fr}} \\ &\equiv m + e_{\mathsf{ct},B} \pmod q, \\ e_{\mathsf{ct},B} &:= e_{\mathsf{ct},A} + p_{\mathsf{ct},A} - \sum_{j=1}^{\ell_{\mathsf{g}}} d_j e_{\mathsf{rk},j} + e_{\mathsf{fr}}. \end{aligned} \tag{14}$$

where the second congruence follows from $\mathbf{Gd} = \mathbf{a} \pmod q$ and the source phase identity (5). The output ciphertext noise $e_{\mathsf{ct},B}$ consists of the inherited source noise $e_{\mathsf{ct},A}$, the pad $p_{\mathsf{ct},A}$, the noise terms $e_{\mathsf{rk},j}$ in the cached rekey, and the noise e_{fr} from the fresh encryption of $p_{\mathsf{ct},A}$.

Let N bound the total number of distinct LP11 encryption instances and Q_{users} the number of registered users, both polynomial. By (37), all key norms are at most $\kappa_{\mathsf{max}} =$

$\underline{\mathsf{Setup}(1^\lambda)}$

1 : Set $T = 1$ and choose n, a prime q, and Gaussian widths $s_{\mathsf{pk}}, s_{\mathsf{err}}, s_{\mathsf{pad}}$ as in (4).
2 : Fix the binary gadget defined above.
3 : Fix a secure PRF family and the bounded Gaussian sampler described above.
4 : **return** pp containing these public parameters and algorithm descriptions.

$\underline{\mathsf{KeyGen}}$

1 : **return** $\mathsf{KeyGen}_{\mathsf{lp}}$

$\underline{\mathsf{ProxyInit}(\mathsf{pp})}$

1 : $\kappa_{\mathsf{pad}} \xleftarrow{\$} \{0,1\}^\lambda$
2 : **return** $\mathsf{st}_{\mathsf{P}} := \kappa_{\mathsf{pad}}$

$\underline{\mathsf{Enc}(\mathsf{pk}, m)}$

1 : $\nu_{\mathsf{src}} \xleftarrow{\$} \{0,1\}^\lambda$
2 : $c \leftarrow \mathsf{Enc}_{\mathsf{lp}}(\mathsf{pk}, m)$
3 : **return** $(0, (c, \nu_{\mathsf{src}}))$

$\underline{\mathsf{Dec}(\mathsf{sk}, (h, \mathsf{ct}))}$

1 : **if** $h \notin \{0,1\}$ or malformed : **return** $\perp$
2 : $\mathbf{s} \leftarrow \mathsf{sk}$; $(c, \sigma) \leftarrow \mathsf{ct}$
3 : **return** $\phi_{\mathbf{s}}(c)$

$\underline{\mathsf{ReKeyGen}(\mathsf{sk}_A, \mathsf{pk}_B)}$

1 : $\mathbf{s}_A \leftarrow \mathsf{sk}_A$
2 : **for** $j \in [\ell_{\mathsf{g}}]$: $K_j \leftarrow \mathsf{Enc}_{\mathsf{lp}}(\mathsf{pk}_B, \langle \mathbf{g}_j, \mathbf{s}_A \rangle)$
3 : **return** $\mathsf{rk}_{A\to B} := (K_j)_{j\in[\ell_{\mathsf{g}}]}$

$\underline{\mathsf{ReEnc}(\mathsf{rk}_{A\to B}, (h, \mathsf{ct}_A), \mathsf{pk}_B; \mathsf{st}_{\mathsf{P}})}$

1 : **if** $h \neq 0$ or the inputs are malformed : **return** $\perp$
2 : $(K_j)_{j\in[\ell_{\mathsf{g}}]} \leftarrow \mathsf{rk}_{A\to B}$
3 : $(c_A, \nu_{\mathsf{src}}) \leftarrow \mathsf{ct}_A$; $(\mathbf{a}, v) \leftarrow c_A$; $\mathbf{d} \leftarrow \mathsf{Decomp}(\mathbf{a})$
4 : $\kappa_{\mathsf{pad}} \leftarrow \mathsf{st}_{\mathsf{P}}$
5 : $p_{\mathsf{ct},A} \leftarrow \mathsf{Sample}_{\mathsf{pad}}\big(F_{\kappa_{\mathsf{pad}}}(\mathtt{src}, \nu_{\mathsf{src}})\big)$
6 : $P \leftarrow \mathsf{Enc}_{\mathsf{lp}}(\mathsf{pk}_B, p_{\mathsf{ct},A} \bmod q)$
7 : $c_B \leftarrow (\mathbf{0}, v) - \sum_{j=1}^{\ell_{\mathsf{g}}} d_j K_j + P$ (13)
8 : **return** $(1, (c_B, \perp))$

Figure 3: Single-hop public-key PRE.

$\lfloor 1.28 n s_{\mathsf{pk}}^2 \rfloor = \Theta(n^2)$ [MS25, Corollary 2] except with probability $Q_{\mathsf{users}} 2^{-2n}$. Conditional on these key bounds, $s_{\mathsf{ct}}(\mathsf{pk}) \le s_{\mathsf{ct,max}} = \sqrt{2\kappa_{\mathsf{max}}}\, s_{\mathsf{err}} = O(n^{3/2}\log^2 n)$. Set $B_{\mathsf{err}} := s_{\mathsf{ct,max}} \log n = O(n^{3/2}\log^3 n)$. At the first generation of each instance, its payload does not affect the noise law and its coins are fresh, so (6) and a union bound give $|e_i| \le B_{\mathsf{err}}$ for all instances except with probability $2Ne^{-\pi(\log n)^2}$.

The source pad bound and $\|\mathbf{d}\|_1 \le \ell_{\mathsf{g}}$ therefore give

$$|e_{\mathsf{ct},B}| \le (\ell_{\mathsf{g}} + 2) B_{\mathsf{err}} + B_{\mathsf{pad}}. \tag{15}$$

Since $\ell_{\mathsf{g}} = O(n \log n)$ and $B_{\mathsf{pad}} = O(\sqrt{n}\log n)$, the right-hand side is $O(n^{5/2}\log^4 n) = o(q)$. Source noise is bounded by B_{err}, so both sources and re-encryptions decode correctly for sufficiently large n, with failure probability at most $2Ne^{-\pi(\log n)^2} + Q_{\mathsf{users}} 2^{-2n} = \mathsf{negl}(n)$.

5.4 Ordered Single-Hop HRA Security

We follow the reverse-rank strategy of [YZD+26]. At each honest user B, Lemma 6 jointly hides its public key and all LP11 encryption instances under it. Figure 4 shows the relevant delegations. The H2C identity below lets us answer corrupted-recipient queries using source NEL.

Proposition 1 (Exact H2C representation)**.** *Consider an admissible H2C response for a nonchallenge source ciphertext* $(0, \mathsf{ct}_A)$*, with* $\mathsf{ct}_A = (c_A, \nu_{\mathsf{src}})$*, LP11 ciphertext* $c_A = (\mathbf{a}, v)$*,*

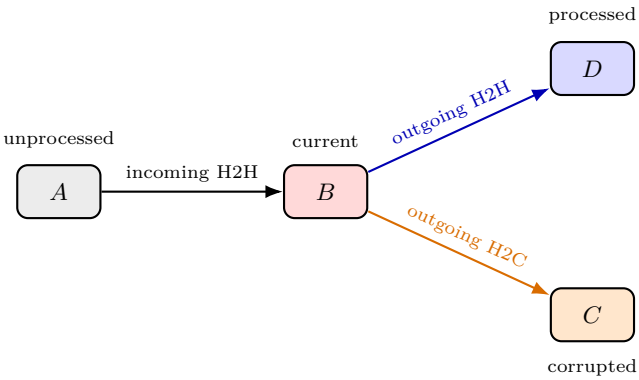

Figure 4: One reverse-rank hybrid step, with $\mathrm{rank}(A) < \mathrm{rank}(B) < \mathrm{rank}(D)$ (user C is corrupted). Outgoing H2H material is already uniform. At B, one joint comparison hides its key, source encryptions, and incoming encryption instances; admissible H2C responses use source NEL.

payload μ_A, and NEL $\ell_{\mathsf{ct},A} := e_{\mathsf{ct},A} + p_{\mathsf{ct},A}$. Let $\mathbf{d} = \mathsf{Decomp}(\mathbf{a})$. Using the same encryption coins as in the real response, define

$$K_j^{(0)} := \mathsf{Enc}_{\mathsf{lp}}(\mathsf{pk}_B, 0; \rho_j), \quad K_j = K_j^{(0)} + (\mathbf{0}, \langle \mathbf{g}_j, \mathbf{s}_A \rangle),$$
$$P^{(0)} := \mathsf{Enc}_{\mathsf{lp}}(\mathsf{pk}_B, 0; \rho_P), \quad P = P^{(0)} + (\mathbf{0}, p_{\mathsf{ct},A}).$$

Then the output LP11 ciphertext satisfies, exactly in $\mathbb{Z}_q^n \times \mathbb{Z}_q$,

$$c_B = (\mathbf{0}, v) - \sum_{j=1}^{\ell_g} d_j K_j + P = (\mathbf{0}, \mu_A + \ell_{\mathsf{ct},A}) - \sum_{j=1}^{\ell_g} d_j K_j^{(0)} + P^{(0)}. \tag{16}$$

Proof. The payload-translation identity for $\mathsf{Enc}_{\mathsf{lp}}$ gives the two decompositions with identical coins. Substitute them into (13). Since $\sum_j d_j \mathbf{g}_j = \mathbf{a} \bmod q$ and $v - \langle \mathbf{a}, \mathbf{s}_A \rangle \equiv \mu_A + e_{\mathsf{ct},A} \pmod q$, the remaining scalar is $\mu_A + e_{\mathsf{ct},A} + p_{\mathsf{ct},A} = \mu_A + \ell_{\mathsf{ct},A}$. □

Since H2C rekeys are never released, the simulator can cache $K_j^{(0)}$ once per H2C delegation pair and sample $P^{(0)}$ afresh for each response. Equation (16) then allows the simulator to answer using the known source message, the NEL, and visible decomposition, without knowing the source secret or pad separately.

Theorem 4 (Ordered single-hop HRA security for discrete public-key PRE)**.** *Use the parameters (4) and the pad sampler specified above. Under Gaussian search-LWE with dimension n, modulus q, and width s_{pk}, and PRF security, the construction of Section 5.2 satisfies, for every PPT adversary $\mathcal{A}$,*

$$\mathsf{Adv}^{\mathsf{ord\text{-}hra}}_{\mathsf{pkpre}}(1^\lambda, \mathcal{A}) = \mathrm{negl}(\lambda). \tag{17}$$

Proof. Fix polynomial bounds Q_{users}, Q_{hon}, Q_{enc}, and Q_{reenc} on registered users, honest users, ordinary encryptions, and re-encryptions, respectively. Including one possible challenge encryption, the total number of LP11 instances under any one honest key is at most

$$M = Q_{\mathsf{enc}} + 1 + \ell_{\mathsf{g}}(Q_{\mathsf{users}} - 1) + Q_{\mathsf{reenc}}. \tag{18}$$

This counts source encryptions, challenge, all incoming cached rekey components, and fresh encryptions of input pads. Write G_X^β for the adversary-visible transcript and final

guess in game X with challenge bit fixed to β, excluding the challenger's private state. Here G_0^β is the real experiment.

By PRF security and the sampling guarantee, replace the pads at distinct source-tag inputs by independent $D_{\mathbb{Z},s_{\mathsf{pad}}}$ samples, preserving repeated accesses. This gives

$$G_0^\beta \approx_c G_1^\beta.$$

In G_2^β, abort on a source-tag collision. Across all users, at most $Q_{\mathsf{enc}}+1$ source tags are sampled independently, so their collision probability is at most $\binom{Q_{\mathsf{enc}}+1}{2}2^{-\lambda}$. This collision probability is negligible, and the tag and norm distributions are unchanged across hybrids. In G_3^β, simulate admissible H2C replies by Proposition 1, keeping the pairwise cache. Hence

$$G_1^\beta \approx_s G_2^\beta \equiv G_3^\beta.$$

Starting from $H_{Q_{\mathsf{hon}}}^\beta := G_3^\beta$, process honest ranks in reverse:

$$H_r^\beta \approx_c H_{r-1}^\beta, \qquad r = Q_{\mathsf{hon}}, \dots, 1. \tag{19}$$

Higher-ranked users have uniform public-key LWE blocks and source LP11 ciphertexts, cached uniform incoming rekey components, and fresh uniform residuals for encryptions of input pads; lower-ranked secrets remain available. Source tags, depths, published norms, and ideal source NEL keep their prescribed distributions. Each reduction fixes its target rank before registration and embeds the target key before revealing it.

For the current user $B = v_r$, apply Lemma 6 once to its public key and at most M LP11 instances, retaining only the NEL of source instances formed with their pads. Discarding the other NEL coordinates preserves computational indistinguishability.

The wrapper can compute all incoming payloads from lower-ranked honest secrets and pads or known corrupted-source state. Outgoing H2H targets have already been processed, so their uniform rekeys and fresh residuals need neither $\mathbf{s}_B$ nor its source pads. Outgoing H2C replies use the known nonchallenge message and source NEL through Proposition 1, with the same pairwise cache. Thus the complete view is online post-processing of the joint lemma without access to the current secret or separate source pads. This proves (19) and preserves the invariant.

Final game. In H_0^β, all honest public-key LWE blocks and source LP11 ciphertexts are uniform, while published norms and ideal NEL values still follow their prescribed joint law. Incoming rekeys under honest targets are cached uniform ciphertexts. Each response to an honest target adds a fresh uniform residual to the visible linear combination, so its output is independent of the challenge bit. H2C replies concern only nonchallenge source records. Challenge-derived outputs are terminal and cannot be converted again. Thus $H_0^0 \equiv H_0^1$ for the visible experiments, including source tags, depths, rejection decisions, and the final guess. There are polynomially many honest ranks and reserved slots, so the preceding comparisons give

$$G_0^0 \approx_c H_0^0 \equiv H_0^1 \approx_c G_0^1. \tag{20}$$

Therefore $\mathsf{Adv}_{\mathsf{pkpre}}^{\mathsf{ord\text{-}hra}}(1^\lambda, \mathcal{A}) = \mathrm{negl}(\lambda)$. □

6 Multi-Hop PRE

We extend the construction of Section 5 to a fixed constant number of hops.

6.1 Multi-Hop Construction

In the multi-hop setting, the proxy needs the pad of a derived ciphertext to re-encrypt it again. We include the required tags and coefficients in the ciphertext header, so the proxy can recompute this pad using its PRF key. The proxy therefore does not need to store a separate pad for each ciphertext.

Reuse the LP11 encryption layer, message space, binary gadget, PRF, and pad sampler of Sections 4.1 and 5.2.

Pads and encryption tags. Source ciphertexts keep their pads. Each cached rekey component and each fresh encryption of the input pad has its own pad, reused whenever that instance is reused. A derived ciphertext receives no additional independent pad. For input LP11 ciphertext $c_A = (\mathbf{a}_A, v_A)$, put $\mathbf{d}_A = \mathsf{Decomp}(\mathbf{a}_A)$. A re-encryption from A to B produces ciphertext noise and a pad given by

$$e_{\mathsf{ct},B} = e_{\mathsf{ct},A} + p_{\mathsf{ct},A} - \sum_j d_{A,j} e_{\mathsf{rk},j} + e_{\mathsf{fr}}. \tag{21}$$

$$p_{\mathsf{ct},B} = -\sum_j d_{A,j} p_{\mathsf{rk},j} + p_{\mathsf{fr}}. \tag{22}$$

Here $e_{\mathsf{rk},j}$ and e_{fr} are the ciphertext noise values of the cached rekey components and the fresh encryption of $p_{\mathsf{ct},A}$, respectively. Their pads are $p_{\mathsf{rk},j}$ and p_{fr}. The pad $p_{\mathsf{ct},B}$ is used at the next hop.

Keep the source tags and pad assignment of (12). For each new cached rekey instance and each fresh encryption of the input pad, independently sample public tags $\nu_{\mathsf{rk}}, \nu_{\mathsf{fr}} \xleftarrow{\$} \{0,1\}^\lambda$. Extend the same domain-separated indexing rule by

$$p_{\mathsf{rk},j} = \mathsf{Sample}_{\mathsf{pad}}\big(F_{\kappa_{\mathsf{pad}}}(\mathtt{rk}, \nu_{\mathsf{rk}}, j)\big), \tag{23}$$

$$p_{\mathsf{fr}} = \mathsf{Sample}_{\mathsf{pad}}\big(F_{\kappa_{\mathsf{pad}}}(\mathtt{fr}, \nu_{\mathsf{fr}})\big). \tag{24}$$

Headers and algorithms. Use the ciphertext format (h, ct) with $\mathsf{ct} = (c, \sigma)$ from Section 5.2. The source header remains $\sigma = \nu_{\mathsf{src}}$. At positive depth, a re-encryption from A to B produces

$$\sigma_B = (\nu_{\mathsf{rk}}, \mathbf{d}_A, \nu_{\mathsf{fr}}). \tag{25}$$

For $h = 0$, $\mathsf{Pad}_{\kappa_{\mathsf{pad}}}(h, \mathsf{ct})$ returns (12). For $h > 0$ and $\sigma = (\nu_r, \mathbf{d}, \nu_f)$, it returns

$$\begin{aligned} \mathsf{Pad}_{\kappa_{\mathsf{pad}}}(h, \mathsf{ct}) = &-\sum_j d_j \mathsf{Sample}_{\mathsf{pad}}\big(F_{\kappa_{\mathsf{pad}}}(\mathtt{rk}, \nu_r, j)\big) \\ &+ \mathsf{Sample}_{\mathsf{pad}}\big(F_{\kappa_{\mathsf{pad}}}(\mathtt{fr}, \nu_f)\big). \end{aligned} \tag{26}$$

Thus the header and the PRF key recover the next pad.

Figure 5 gives the algorithms.

6.2 Ordered Multi-Hop HRA Security

Let $\mathsf{Exp}_{\Pi}^{\mathsf{ord\text{-}hra},t}$ denote Fig. 1 with $T = t$ and the H2H order restriction of Section 5.1, on both rekey release and transformation.

Theorem 5 (Ordered constant-hop HRA security)**.** *Fix a constant $t \geq 1$ and use the parameters (4) and the pad sampler of Section 5.2. Under Gaussian search-LWE with dimension n, modulus q, and width s_{pk}, and PRF security, the construction in Fig. 5 satisfies $\mathsf{Adv}_{\Pi}^{\mathsf{ord\text{-}hra},t}(1^\lambda, \mathcal{A}) = \mathsf{negl}(\lambda)$ for every PPT adversary $\mathcal{A}$.*

$\underline{\mathsf{Setup}(1^\lambda)}$

1 : Fix a constant hop bound $t \geq 1$ and the LP11, gadget, PRF, and sampler parameters above, using (4).
2 : **return** pp containing these public parameters and algorithm descriptions.

$\underline{\mathsf{KeyGen}}$

1 : **return** $\mathsf{KeyGen}_{\mathsf{lp}}$

$\underline{\mathsf{ProxyInit}(\mathsf{pp})}$

1 : $\kappa_{\mathsf{pad}} \xleftarrow{\$} \{0,1\}^\lambda$
2 : **return** $\mathsf{st}_{\mathsf{P}} := \kappa_{\mathsf{pad}}$

$\underline{\mathsf{Enc}(\mathsf{pk}, m)}$

1 : $\nu_{\mathsf{src}} \xleftarrow{\$} \{0,1\}^\lambda$
2 : $c \leftarrow \mathsf{Enc}_{\mathsf{lp}}(\mathsf{pk}, m)$
3 : **return** $(0, (c, \nu_{\mathsf{src}}))$

$\underline{\mathsf{Dec}(\mathsf{sk}, (h, \mathsf{ct}))}$

1 : **if** $h \notin \{0, \ldots, t\}$: **return** $\perp$
2 : $(c, \sigma) \leftarrow \mathsf{ct}$; $\mathbf{s} \leftarrow \mathsf{sk}$
3 : **return** $\phi_{\mathbf{s}}(c)$

$\underline{\mathsf{ReKeyGen}(\mathsf{sk}_A, \mathsf{pk}_B)}$

1 : $\mathbf{s}_A \leftarrow \mathsf{sk}_A$
2 : **for** $j \in [\ell_{\mathsf{g}}]$: $K_j \leftarrow \mathsf{Enc}_{\mathsf{lp}}(\mathsf{pk}_B, \langle \mathbf{g}_j, \mathbf{s}_A \rangle)$
3 : $\nu_{\mathsf{rk}} \xleftarrow{\$} \{0,1\}^\lambda$
4 : **return** $\mathsf{rk}_{A\to B} := (\nu_{\mathsf{rk}}, (K_j)_{j\in[\ell_{\mathsf{g}}]})$

$\underline{\mathsf{ReEnc}(\mathsf{rk}_{A\to B}, (h, \mathsf{ct}_A), \mathsf{pk}_B; \mathsf{st}_{\mathsf{P}})}$

1 : **if** $h \notin \{0, \ldots, t-1\}$ or the inputs are malformed : **return** $\perp$
2 : $(\nu_{\mathsf{rk}}, (K_j)_{j\in[\ell_{\mathsf{g}}]}) \leftarrow \mathsf{rk}_{A\to B}$; $\kappa_{\mathsf{pad}} \leftarrow \mathsf{st}_{\mathsf{P}}$
3 : $p_{\mathsf{ct},A} \leftarrow \mathsf{Pad}_{\kappa_{\mathsf{pad}}}(h, \mathsf{ct}_A)$
4 : $(c_A, \sigma_A) \leftarrow \mathsf{ct}_A$; $(\mathbf{a}_A, v_A) \leftarrow c_A$; $\mathbf{d}_A \leftarrow \mathsf{Decomp}(\mathbf{a}_A)$
5 : $\nu_{\mathsf{fr}} \xleftarrow{\$} \{0,1\}^\lambda$
6 : $P \leftarrow \mathsf{Enc}_{\mathsf{lp}}(\mathsf{pk}_B, p_{\mathsf{ct},A} \bmod q)$
7 : $c_B \leftarrow (\mathbf{0}, v_A) - \sum_{j=1}^{\ell_{\mathsf{g}}} d_{A,j} K_j + P$
8 : $\sigma_B \leftarrow (\nu_{\mathsf{rk}}, \mathbf{d}_A, \nu_{\mathsf{fr}})$; **return** $(h+1, (c_B, \sigma_B))$

Figure 5: Multi-hop public-key PRE with pad-recovery headers.

Write $\ell_i = e_i + p_i$ for each LP11 instance and $\ell_{\mathsf{ct},A} = e_{\mathsf{ct},A} + p_{\mathsf{ct},A}$ for each record. Equations (21)–(22) give

$$\ell_{\mathsf{ct},B} = \ell_{\mathsf{ct},A} - \sum_j d_{A,j}\ell_{\mathsf{rk},j} + \ell_{\mathsf{fr}}. \tag{27}$$

The proof follows the reverse-rank argument of Theorem 4. Joint hiding applies to source encryptions, cached rekey components, and fresh encryptions of input pads; derived ciphertexts use the NEL recurrence above. Appendix D gives the proof, focusing on this invariant and the online simulation for derived inputs.

6.3 Multi-Hop Correctness

Suppose the ciphertext noise and pad of every LP11 instance in an execution satisfy $|e_i| \le B_{\mathsf{err}}$ and $|p_i| \le B_{\mathsf{pad}}$. A source record has $|e_{\mathsf{ct},0}| \le B_{\mathsf{err}}$ and $|p_{\mathsf{ct},0}| \le B_{\mathsf{pad}}$. Since $\|\mathbf{d}\|_1 \le \ell_{\mathsf{g}}$, Equation (22) gives $|p_{\mathsf{ct},h}| \le (\ell_{\mathsf{g}}+1)B_{\mathsf{pad}}$ for every derived record. By (21), each hop increases the noise magnitude by at most its input pad and $(\ell_{\mathsf{g}}+1)B_{\mathsf{err}}$. Consequently, for $1 \le h \le t$,

$$|e_{\mathsf{ct},h}| \le B_{\mathsf{err}} + B_{\mathsf{pad}} + h(\ell_{\mathsf{g}}+1)B_{\mathsf{err}} + (h-1)(\ell_{\mathsf{g}}+1)B_{\mathsf{pad}}. \tag{28}$$

The fresh-instance tail argument in Section 5.3 bounds every LP11 noise by $B_{\mathsf{err}} = O(n^{3/2}\log^3 n)$ except with negligible probability, while sampled pads are bounded by $B_{\mathsf{pad}} = O(\sqrt{n}\log n)$. Substituting these bounds and $\ell_{\mathsf{g}} = O(n\log n)$ into (28) gives $|e_{\mathsf{ct},h}| = O(n^{5/2}\log^4 n) = o(q)$ for every $h \le t$ with fixed t. Thus all outputs decode correctly, with the same negligible failure bound as in Section 5.3.

7 Removing Delegation-Graph Restrictions by Key Separation

We remove the honest-user order in Theorems 4 and 5 using independent keys at each ciphertext depth. Base rekeys advance one level, allowing cycles in the user delegation graph while keeping key dependencies acyclic. The compiler applies to any polynomial hop bound supported by an ordered HRA-secure base scheme.

7.1 A Compiler for t Hops

Let Π be a t-hop PRE scheme. We use the HRA experiment of Section 2.3 and its ordered version from Section 5.1.

Construction. Define $\Pi^{t\text{-hop}}$ by running base setup and proxy initialisation once. Each user A generates $t+1$ independent base key pairs, with composite keys

$$\mathsf{pk}_A = (\mathsf{pk}_{A,0}, \dots, \mathsf{pk}_{A,t}), \qquad \mathsf{sk}_A = (\mathsf{sk}_{A,0}, \dots, \mathsf{sk}_{A,t}).$$

Use the base ciphertext (h, ct) unchanged. Set

$$\mathsf{Enc}^{t\text{-hop}}(\mathsf{pk}_A, m) = \mathsf{Enc}(\mathsf{pk}_{A,0}, m),$$
$$\mathsf{Dec}^{t\text{-hop}}(\mathsf{sk}_A, (h, \mathsf{ct})) = \mathsf{Dec}(\mathsf{sk}_{A,h}, (h, \mathsf{ct})).$$

A rekey is the tuple $\mathsf{rk}^{t\text{-hop}}_{A\to B} = (R^{(h)}_{A,B})_{0\le h<t}$, where

$$R^{(h)}_{A,B} \leftarrow \mathsf{ReKeyGen}(\mathsf{sk}_{A,h}, \mathsf{pk}_{B,h+1}). \tag{29}$$

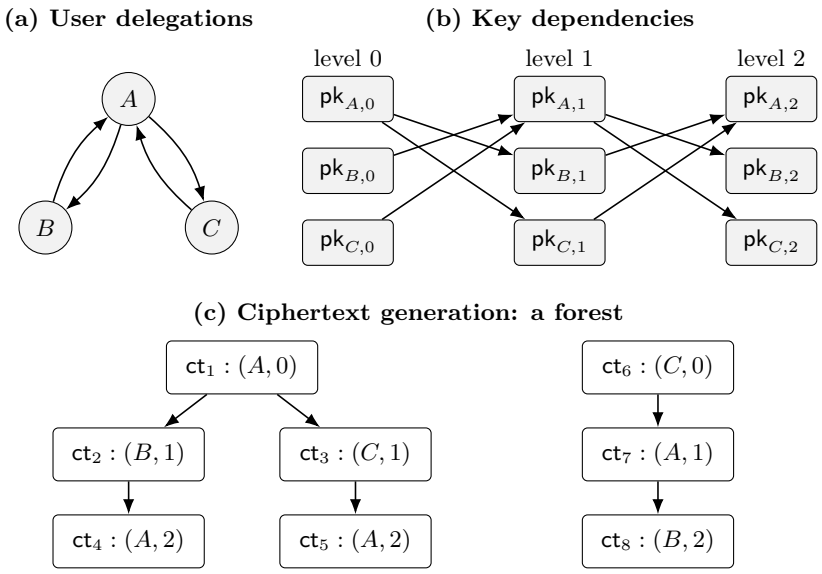

Figure 6: A cyclic user graph, an acyclic key graph, and a ciphertext forest ($t = 2$). (a) Users may delegate in both directions. (b) Every base rekey goes from level h to $h + 1$; for one hop, keep only levels 0 and 1. (c) A possible execution with two source records. A node $\mathsf{ct}_i : (U, h)$ denotes record i containing (h, ct_i) and owned by U. Records 4 and 5 both use $\mathsf{pk}_{A,2}$ but remain separate vertices.

For $h < t$, re-encryption returns the result of

$$\mathsf{ReEnc}(R^{(h)}_{A,B}, (h, \mathsf{ct}), \mathsf{pk}_{B,h+1}; \mathsf{st}_\mathsf{P}). \tag{30}$$

The output is $(h + 1, \mathsf{ct}')$ or $\perp$, as in the base scheme.

Every compiled operation is a base operation on the same tagged ciphertext and shared proxy state, so the base correctness bound carries over at the enlarged user and query bounds.

Lemma 7 (Key dependencies and ciphertext generation)**.** *The key-dependency graph of $\Pi^{t\text{-hop}}$ has one vertex per component key pair and one edge $A^{(h)} \to B^{(h+1)}$ per base rekey. It is a DAG with directed paths of length at most t. The ciphertext-generation graph has one vertex per record and one edge per successful transformation. It is a forest rooted at source records, with depth at most t.*

Proof. Every edge in the key-dependency graph increases the level by one. Hence ordering the keys by level puts every edge forward. Each successful transformation creates a new record with exactly one parent, at the next depth. Thus the descendants of each source record form a tree, even when a record is used repeatedly or two ciphertext encodings coincide. □

Figure 6 shows all three graphs for $t = 2$ and users A, B, C.

7.2 HRA Security and Instantiation

Theorem 6 (HRA security of the t-hop compiler)**.** *Let Π be a t-hop PRE scheme with polynomially bounded t. If Π is ordered t-hop HRA secure, then $\Pi^{t\text{-hop}}$ is t-hop HRA secure without an honest-user order.*

The proof is given in Appendix E.

Applying the compiler. For the schemes in Sections 5 and 6, the shared proxy state is their single private PRF seed. The compiler uses the existing pad algorithms unchanged. Applying Theorem 6 to these schemes gives the following corollary.

Corollary 1 (HRA security without an honest-user order)**.** *Under the assumptions of Theorem 4, its key-separation compilation is single-hop HRA secure. Under the assumptions of Theorem 5 and the parameters* (4)*, the key-separation compilation of Section 6 is HRA secure and correct for any fixed constant hop bound t.*

Cost. Each user stores $t+1$ base key pairs, and each delegation uses t base rekeys. The reduction multiplies the honest and corrupted registration bounds by $t+1$ and the rekey-release bound by at most t; other oracle query counts do not increase. Ciphertext sizes are unchanged. Encryption, decryption, and each transformation invoke the corresponding base algorithm once, with one shared proxy state. Key separation adds no ciphertext noise; the base security and correctness bounds are evaluated at the larger component counts.

AI Tool Use Disclosure

OpenAI's ChatGPT and Codex were used to assist with language editing, LaTeX preparation, literature checking, and proof checking. The authors have reviewed and verified the proofs and take full responsibility for the accuracy and integrity of the entire manuscript.

References

[ACPS09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009*, pages 595–618, Berlin, Heidelberg, 2009. Springer. doi:10.1007/978-3-642-03356-8_35.

[AFGH06] Giuseppe Ateniese, Kevin Fu, Matthew Green, and Susan Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Trans. Inf. Syst. Secur.*, 9(1):1–30, February 2006. URL: https://dl.acm.org/doi/10.1145/1127345.1127346, doi:10.1145/1127345.1127346.

[BBS98] Matt Blaze, Gerrit Bleumer, and Martin Strauss. Divertible protocols and atomic proxy cryptography. In Kaisa Nyberg, editor, *Advances in Cryptology — EUROCRYPT'98*, pages 127–144, Berlin, Heidelberg, 1998. Springer. doi:10.1007/BFb0054122.

[BV11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient Fully Homomorphic Encryption from (Standard) LWE. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 97–106, October 2011. URL: https://ieeexplore.ieee.org/document/6108154, doi:10.1109/FOCS.2011.12.

[CCG+25] Aloni Cohen, David Bruce Cousins, Nicholas Genise, Erik Kline, Yuriy Polyakov, and Saraswathy RV. HRA-Secure Homomorphic Lattice-Based Proxy Re-Encryption with Tight Security. *CiC*, 2(1):1–30, April 2025. URL: https://cic.iacr.org/p/2/1/22, doi:10.62056/ab0l5wol7.

[Coh19] Aloni Cohen. What About Bob? The Inadequacy of CPA Security for Proxy Reencryption. In Dongdai Lin and Kazue Sako, editors, *Public-Key Cryptography – PKC 2019*, pages 287–316, Cham, 2019. Springer International Publishing. doi:10.1007/978-3-030-17259-6_10.

[DDLM19] Alex Davidson, Amit Deo, Ela Lee, and Keith Martin. Strong Post-Compromise Secure Proxy Re-Encryption. In Julian Jang-Jaccard and Fuchun Guo, editors, *Information Security and Privacy*, pages 58–77, Cham, 2019. Springer International Publishing. doi:10.1007/978-3-030-21548-4_4.

[FKKP19] Georg Fuchsbauer, Chethan Kamath, Karen Klein, and Krzysztof Pietrzak. Adaptively Secure Proxy Re-encryption. In Dongdai Lin and Kazue Sako, editors, *Public-Key Cryptography – PKC 2019*, pages 317–346, Cham, 2019. Springer International Publishing. doi:10.1007/978-3-030-17259-6_11.

[FL17] Xiong Fan and Feng-Hao Liu. Proxy Re-Encryption and Re-Signatures from Lattices, 2017. URL: https://eprint.iacr.org/2017/456.

[FL19] Xiong Fan and Feng-Hao Liu. Proxy Re-Encryption and Re-Signatures from Lattices. In Robert H. Deng, Valérie Gauthier-Umaña, Martín Ochoa, and Moti Yung, editors, *Applied Cryptography and Network Security*, pages 363–382, Cham, 2019. Springer International Publishing. doi:10.1007/978-3-030-21568-2_18.

[GMPW20] Nicholas Genise, Daniele Micciancio, Chris Peikert, and Michael Walter. Improved Discrete Gaussian and Subgaussian Analysis for Lattice Cryptography. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *Public-Key Cryptography – PKC 2020*, pages 623–651, Cham, 2020. Springer International Publishing. Full version: Cryptology ePrint Archive, Report 2020/337. URL: https://eprint.iacr.org/2020/337, doi:10.1007/978-3-030-45374-9_21.

[Kir14] Elena Kirshanova. Proxy Re-encryption from Lattices. In Hugo Krawczyk, editor, *Public-Key Cryptography – PKC 2014*, volume 8383, pages 77–94. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014. URL: http://link.springer.com/10.1007/978-3-642-54631-0_5, doi:10.1007/978-3-642-54631-0_5.

[LP11] Richard Lindner and Chris Peikert. Better Key Sizes (and Attacks) for LWE-Based Encryption. In Aggelos Kiayias, editor, *Topics in Cryptology – CT-RSA 2011*, pages 319–339, Berlin, Heidelberg, 2011. Springer. doi:10.1007/978-3-642-19074-2_21.

[LSW25] Russell Lai, Monisha Swarnakar, and Ivy Woo. Leaky LWE: Learning with Errors with Semi-Adaptive Secret- and Error-Leakage. *IACR CiC*, 2(3):cc2–3–35, October 2025. URL: https://cic.iacr.org/p/2/3/21, doi:10.62056/ah89ksuc2.

[MM11] Daniele Micciancio and Petros Mol. Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 465–484, Berlin, Heidelberg, 2011. Springer. doi:10.1007/978-3-642-22792-9_26.

[MP11] Daniele Micciancio and Chris Peikert. Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller, 2011. URL: https://eprint.iacr.org/2011/501.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, pages 700–718, Berlin, Heidelberg, 2012. Springer. doi:10.1007/978-3-642-29011-4_41.

[MP13] Daniele Micciancio and Chris Peikert. Hardness of SIS and LWE with small parameters. In *Advances in Cryptology – CRYPTO 2013, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 21–39. Springer, 2013. Full version: Cryptology ePrint Archive, Paper 2013/069. URL: https://eprint.iacr.org/2013/069, doi:10.1007/978-3-642-40041-4_2.

[MR07] Daniele Micciancio and Oded Regev. Worst-Case to Average-Case Reductions Based on Gaussian Measures. *SIAM J. Comput.*, 37(1):267–302, January 2007. URL: https://epubs.siam.org/doi/abs/10.1137/S0097539705447360, doi:10.1137/S0097539705447360.

[MS25] Daniele Micciancio and Adam Suhl. Simulation-Secure Threshold PKE from LWE with Polynomial Modulus. *IACR CiC*, 1(4):cc1–3–24, January 2025. URL: https://cic.iacr.org/p/1/4/2, doi:10.62056/a0zogy4e-.

[PRSV17] Yuriy Polyakov, Kurt Rohloff, Gyana Sahu, and Vinod Vaikuntanathan. Fast Proxy Re-Encryption for Publish/Subscribe Systems. *ACM Trans. Priv. Secur.*, 20(4):14:1–14:31, September 2017. URL: https://dl.acm.org/doi/10.1145/3128607, doi:10.1145/3128607.

[SDDR21] Willy Susilo, Priyanka Dutta, Dung Hoang Duong, and Partha Sarathi Roy. Lattice-Based HRA-secure Attribute-Based Proxy Re-Encryption in Standard Model. In Elisa Bertino, Haya Shulman, and Michael Waidner, editors, *Computer Security – ESORICS 2021*, pages 169–191, Cham, 2021. Springer International Publishing. doi:10.1007/978-3-030-88428-4_9.

[YWW+25a] Lisha Yao, Jian Weng, Pengfei Wu, Guofeng Tang, Haiyang Xue, Guomin Yang, and Robert H. Deng. Attribute-Based Conditional PRE: A Novel Construction from LWE for Cloud Data-Sharing. In *2025 IEEE 24th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 793–804, Guiyang, China, November 2025. IEEE. URL: https://ieeexplore.ieee.org/document/11354840/, doi:10.1109/Trustcom66490.2025.00093.

[YWW+25b] Lisha Yao, Jian Weng, Pengfei Wu, Guofeng Tang, Guomin Yang, Haiyang Xue, and Robert H. Deng. Conditional Attribute-Based PRE: Definition and Construction from LWE. In Sang Kil Cha and Jeongeun Park, editors, *Information Security*, pages 131–151, Cham, 2025. Springer Nature Switzerland. doi:10.1007/978-3-032-08124-7_7.

[YZD+26] Haotian Yin, Jie Zhang, Yuji Dong, Dominik Wojtczak, and Eng Gee Lim. Single-Hop HRA-Secure Owner-Encrypted Lattice-Based Proxy Re-Encryption without Statistical Noise Flooding, 2026. URL: https://eprint.iacr.org/2026/1113.

[YZL+25] Haotian Yin, Jie Zhang, Wanxin Li, Yuji Dong, Eng Gee Lim, and Dominik Wojtczak. Collusion-Safe Proxy Re-Encryption, 2025. URL: https://eprint.iacr.org/2025/1423.

[ZCA+25] Feixiang Zhao, Shixin Chen, Man Ho Au, Jian Weng, Huaxiong Wang, and Jian Guo. Bootstrappable Fully Homomorphic Attribute-Based Encryption

with Unbounded Circuit Depth. In Goichiro Hanaoka and Bo-Yin Yang, editors, *Advances in Cryptology – ASIACRYPT 2025*, pages 230–262, Singapore, 2025. Springer Nature. doi:10.1007/978-981-95-5122-4_8.

[ZCL26] Mengdi Zhao, Huiyan Chen, and Xi Lin. Lattice-based puncturable attribute-based proxy re-encryption scheme in cloud computing. *Journal of Information Security and Applications*, 97:104325, March 2026. URL: https://linkinghub.elsevier.com/retrieve/pii/S2214212625003618, doi:10.1016/j.jisa.2025.104325.

[ZJL+25] Tianqiao Zhang, Mingming Jiang, Fucai Luo, Yuyan Guo, and Jinqiu Hou. HRA-Secure Puncturable Attribute-Based Proxy Re-Encryption from Lattices for Secure Cloud Sharing, 2025. URL: https://eprint.iacr.org/2025/2105.

[ZLH24] Yunxiao Zhou, Shengli Liu, and Shuai Han. Multi-hop Fine-Grained Proxy Re-encryption. In Qiang Tang and Vanessa Teague, editors, *Public-Key Cryptography – PKC 2024*, pages 161–192, Cham, 2024. Springer Nature Switzerland. doi:10.1007/978-3-031-57728-4_6.

[ZLHZ23] Yunxiao Zhou, Shengli Liu, Shuai Han, and Haibin Zhang. Fine-Grained Proxy Re-encryption: Definitions and Constructions from LWE. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology – ASIACRYPT 2023*, pages 199–231, Singapore, 2023. Springer Nature. doi:10.1007/978-981-99-8736-8_7.

[ZWW23] Yu Zhang, Yang Wang, and Mingqiang Wang. Improved AB-CPREs with HRA Security. In *2023 International Conference on Data Security and Privacy Protection (DSPP)*, pages 223–231, Xi'an, October 2023. IEEE. URL: https://ieeexplore.ieee.org/document/10404893, doi:10.1109/DSPP58763.2023.10404893.

[ZWX+24a] Feixiang Zhao, Jian Weng, Wenli Xie, Lin Hou, and Ming Li. Time-based attribute-based proxy re-encryption with decryption key update. *Des. Codes Cryptogr.*, 92(12):4099–4129, December 2024. doi:10.1007/s10623-024-01467-x.

[ZWX+24b] Feixiang Zhao, Jian Weng, Wenli Xie, Ming Li, and Jiasi Weng. HRA-secure attribute-based threshold proxy re-encryption from lattices. *Information Sciences*, 655:119900, January 2024. URL: https://www.sciencedirect.com/science/article/pii/S0020025523014858, doi:10.1016/j.ins.2023.119900.

A Noise-Averaged Reconstruction Attack

Both the ACNS 2019 construction and the ePrint full version (Section 4) display the product $\mathbf{b}^\top \mathsf{rk}_{A\to B}$ without fresh error, while their expanded expression includes independently sampled level-2 error [FL19, FL17]. We analyse this noise-bearing interpretation under explicit modulus-to-noise and repetition conditions, or we can use the same functional-reconstruction directly as the attack on Susilo et al.'s scheme [SDDR21].

A.1 Re-encryption matrix

Let $n = \mathsf{poly}(\lambda)$ be the lattice dimension, let $q = \mathsf{poly}(n)$ be the ciphertext modulus, and let $k = O(\log q)$ be the gadget-width parameter, so that the underlying gadget matrix has nk columns. In Fan–Liu's notation, a source ciphertext has the form $\mathsf{ct} = (u, \mathbf{b}, 1)$, where u is its public tag and $\mathbf{b} \in \mathbb{Z}_{2q}^{3nk}$; re-encryption preserves u and changes the level to 2.

For a fixed delegation from user A to user B, write the re-encryption matrix as

$$\mathsf{rk}_{A\to B} := \begin{bmatrix} \mathbf{I}_{nk} & \mathbf{X}_{01} & \mathbf{X}_{02} \\ \mathbf{0} & \mathbf{X}_{11} & \mathbf{X}_{12} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{nk} \end{bmatrix} \in \mathbb{Z}^{3nk\times 3nk},$$

where every $\mathbf{X}_{ij}$ is an $nk \times nk$ integer matrix. Honest re-encryption can be written as

$$(\mathbf{b}')^\top = \mathbf{b}^\top \mathsf{rk}_{A\to B} + \mathbf{e}_{\mathsf{re}}^\top \pmod{2q}, \qquad \mathbf{e}_{\mathsf{re}} \leftarrow D_{\mathbb{Z}^{3nk},s_0},$$

where s_0 denotes the public error width (called s' in the conference version). The proxy outputs $(u, \mathbf{b}', 2)$. Consequently, for any fixed source ciphertext, repeated re-encryptions give independent noisy observations of the same deterministic vector $\mathbf{b}^\top \mathsf{rk}_{A\to B} \bmod 2q$.

A.2 Effective fixed linear map

Fix an honest source user A, a corrupted recipient B, and the corresponding fixed rekey $\mathsf{rk}_{A\to B}$. Parse each honest source ciphertext $\mathsf{ct}_i = (u_i, \mathbf{b}_i, 1)$ as $\mathbf{b}_i^\top = [\,\mathbf{b}_{i,0}^\top \mid \mathbf{b}_{i,1}^\top \mid \mathbf{b}_{i,2}^\top\,] \in \mathbb{Z}_{2q}^{3nk}$. Define the effective hidden linear map and its public input vector by

$$\mathbf{T}_{A\to B} := \begin{bmatrix} \mathbf{X}_{01} & \mathbf{X}_{02} \\ \mathbf{X}_{11} & \mathbf{X}_{12} \end{bmatrix} \in \mathbb{Z}^{2nk\times 2nk}, \qquad \mathbf{z}_i^\top := [\,\mathbf{b}_{i,0}^\top \mid \mathbf{b}_{i,1}^\top\,] \in \mathbb{Z}_{2q}^{2nk}.$$

Let $\mathbf{y}_i^{(\nu)}$ be the vector component of the ν-th honest re-encryption of ct_i. Write it in three blocks:

$$(\mathbf{y}_i^{(\nu)})^\top = [\,(\mathbf{y}_{i,0}^{(\nu)})^\top \mid (\mathbf{y}_{i,1}^{(\nu)})^\top \mid (\mathbf{y}_{i,2}^{(\nu)})^\top\,].$$

The adversary computes the adjusted response

$$(\mathbf{w}_i^{(\nu)})^\top := \left[\, (\mathbf{y}_{i,1}^{(\nu)})^\top \;\middle|\; (\mathbf{y}_{i,2}^{(\nu)})^\top - \mathbf{b}_{i,2}^\top \,\right].$$

By the block structure of the rekey,

$$(\mathbf{w}_i^{(\nu)})^\top = \mathbf{z}_i^\top \mathbf{T}_{A\to B} + (\mathbf{e}_i^{(\nu)})^\top \pmod{2q},$$

where $\mathbf{e}_i^{(\nu)} \in \mathbb{Z}^{2nk}$ contains the second and third blocks of the fresh error.

The adversary obtains N_{samp} independently generated honest source ciphertexts under pk_A and requests N_{rep} honest re-encryptions of each one under the same rekey. These are legal nonchallenge queries and may be completed before the challenge is issued.

A.3 Modular lifting

For each i, the adversary aligns the modular responses to a common integer lift before averaging. Let $\widehat{\mathbf{w}}_i^{(1)} \in \{0, \dots, 2q-1\}^{2nk}$ be the canonical lift of the first response, and let $\langle\cdot\rangle_{2q}$ denote the coordinate-wise centred representative in $(-q, q]$. For every $\nu \in [N_{\mathsf{rep}}]$, set

$$\widehat{\mathbf{w}}_i^{(\nu)} := \widehat{\mathbf{w}}_i^{(1)} + \left\langle \mathbf{w}_i^{(\nu)} - \mathbf{w}_i^{(1)} \right\rangle_{2q}. \tag{31}$$

Let $\mathbf{v}_i^\top := \mathbf{z}_i^\top \mathbf{T}_{A\to B} \bmod 2q$. If $\|\mathbf{e}_i^{(\nu)} - \mathbf{e}_i^{(1)}\|_\infty < q$ for every ν, then each centred modular difference in (31) is exactly the corresponding integer error difference. Hence all repeated responses are aligned to one common integer lift: there exists a lift $\widehat{\mathbf{v}}_i$ of $\mathbf{v}_i$ such that $\widehat{\mathbf{w}}_i^{(\nu)} = \widehat{\mathbf{v}}_i + \mathbf{e}_i^{(\nu)}$ for every ν. Let $\mathsf{Good}_{\mathsf{lift}}$ denote the event that the error-difference condition holds for every sample i and repetition ν.

A.4 Gaussian noise averaging

After aligning the responses, the adversary computes

$$\widetilde{\mathbf{v}}_i := \left\lfloor \left\lfloor \frac{1}{N_{\mathsf{rep}}} \sum_{\nu=1}^{N_{\mathsf{rep}}} \widehat{\mathbf{w}}_i^{(\nu)} \right\rceil \right\rceil_{2q}. \tag{32}$$

Whenever $\left\| N_{\mathsf{rep}}^{-1} \sum_{\nu=1}^{N_{\mathsf{rep}}} \mathbf{e}_i^{(\nu)} \right\|_\infty < 1/2$, coordinate-wise rounding in (32) gives $\widetilde{\mathbf{v}}_i = \mathbf{v}_i$. Let $\mathsf{Good}_{\mathsf{avg}}$ denote the event that this mean-error bound holds for every sample i.

A.5 CRT-based functional reconstruction over $\mathbb{Z}_{2q}$

The adversary stacks the recovered pairs into matrices $\mathbf{Z}$ and $\mathbf{V}$ whose i-th rows are $\mathbf{z}_i^\top$ and $\widetilde{\mathbf{v}}_i^\top$, respectively. When noise removal succeeds for every sample,

$$\mathbf{V} = \mathbf{Z}\mathbf{T}_{A\to B} \pmod{2q}.$$

After receiving the challenge ciphertext $\mathsf{ct}^* = (u^*, \mathbf{b}^*, 1)$, parse $(\mathbf{b}^*)^\top = [\,(\mathbf{b}_0^*)^\top \mid (\mathbf{b}_1^*)^\top \mid (\mathbf{b}_2^*)^\top\,]$ and define $(\mathbf{z}^*)^\top := [\,(\mathbf{b}_0^*)^\top \mid (\mathbf{b}_1^*)^\top\,]$.

Fix the honest source user A before preprocessing and condition on the public parameters and its public key pk_A. The vectors $\mathbf{z}_1, \dots, \mathbf{z}_{N_{\mathsf{samp}}}$ obtained from independent honest source encryptions and the challenge vector $\mathbf{z}^*$ are then independent and identically distributed.

Because q is an odd prime, $\mathbb{Z}_{2q} \cong \mathbb{F}_2 \times \mathbb{F}_q$. For $p \in \{2, q\}$, let $\mathsf{Good}_{\mathsf{span},p}$ be the event that $\mathbf{z}^* \bmod p$ lies in the row span of $\mathbf{Z} \bmod p$, and set $\mathsf{Good}_{\mathsf{span}} := \mathsf{Good}_{\mathsf{span},2} \cap \mathsf{Good}_{\mathsf{span},q}$. This event is equivalent to the existence of $\mathbf{g}_{\mathsf{rec}} \in \mathbb{Z}_{2q}^{N_{\mathsf{samp}}}$ satisfying $(\mathbf{z}^*)^\top = \mathbf{g}_{\mathsf{rec}}^\top \mathbf{Z} \bmod 2q$.

The adversary finds such a vector efficiently by solving the corresponding linear systems over $\mathbb{F}_2$ and $\mathbb{F}_q$, then combining the solutions coordinate-wise via the Chinese remainder theorem. If such coefficients exist, then

$$(\mathbf{v}^*)^\top := \mathbf{g}_{\mathsf{rec}}^\top \mathbf{V} = (\mathbf{z}^*)^\top \mathbf{T}_{A\to B} \pmod{2q}.$$

Writing $(\mathbf{v}^*)^\top = [\,(\mathbf{v}_1^*)^\top \mid (\mathbf{v}_2^*)^\top\,]$, the deterministic part of the challenge re-encryption is

$$(\overline{\mathbf{b}}'^*)^\top := \left[\, (\mathbf{b}_0^*)^\top \;\middle|\; (\mathbf{v}_1^*)^\top \;\middle|\; (\mathbf{v}_2^*)^\top + (\mathbf{b}_2^*)^\top \,\right] \pmod{2q}.$$

The adversary restores the honest re-encryption distribution by sampling

$$\widehat{\mathbf{e}}_{\mathsf{re}}^* \leftarrow D_{\mathbb{Z}^{3nk}, s_0}, \qquad (\mathbf{b}'^*)^\top := (\overline{\mathbf{b}}'^*)^\top + (\widehat{\mathbf{e}}_{\mathsf{re}}^*)^\top \pmod{2q}.$$

It forms $\mathsf{ct}'^* := (u^*, \mathbf{b}'^*, 2)$ and decrypts it using the corrupted secret key of user B. When reconstruction succeeds, the recovered deterministic part and fresh error give the honest re-encryption distribution, so ordinary re-encryption correctness applies.

A.6 Sample complexity, tail bounds, and repetition budget

The expanded re-encryption formula uses the public level-2 error distribution. The conference parameter paragraph specifies $s_0 = s' = s\sqrt{m}$, where $m = O(nk)$ and $s = \omega(\sqrt{\log n})\sqrt{m}$ [FL19, Section 4, p. 377]. The full version's Section 4.2 does not separately specify s_0. The notation $\omega(\sqrt{\log n})$ alone does not impose a polynomial upper bound. We therefore state the attack under explicit assumptions on s_0 and q:

$$\varepsilon_{\mathsf{FL}}(\lambda) := \Pr_{e \leftarrow D_{\mathbb{Z},s_0}}[|e| \geq q/2] = \mathsf{negl}(\lambda). \tag{33}$$

For independent $e, e' \leftarrow D_{\mathbb{Z},s_0}$, the event $|e - e'| \geq q$ implies $|e| \geq q/2$ or $|e'| \geq q/2$. Hence

$$\Pr[|e - e'| \geq q] \leq 2\varepsilon_{\mathsf{FL}}(\lambda) = \mathsf{negl}(\lambda). \tag{34}$$

The threshold q is exactly what is needed for lifting modulo $2q$: whenever $|e - e'| < q$, the centred representative modulo $2q$ equals the true integer difference. A sufficient regime for Equation (33) is $q/s_0 = \omega(\sqrt{\log \lambda})$, since the discrete Gaussian tail is at most $2\exp(-\pi q^2/(4s_0^2))$.

Let $d_{\mathsf{rec}} := 2nk$ and set $N_{\mathsf{samp}} := 4d_{\mathsf{rec}}$. Finally, set

$$N_{\mathsf{rep}} := \left\lceil \frac{4s_0^2}{\pi}\big(\lambda \ln 2 + \ln(2N_{\mathsf{samp}}d_{\mathsf{rec}})\big) \right\rceil. \tag{35}$$

A.7 Attack guarantee

Theorem 7 (Noise-averaged attack on tbPRE)**.** *Consider the Fan–Liu tag-based CCA experiment [FL17, Section 3.2] specified in Section 3.2, under the noise-bearing interpretation of re-encryption. Assume that q is an odd prime, $d_{\mathsf{rec}} = \mathsf{poly}(\lambda)$, and $s_0 = \mathsf{poly}(\lambda)$. Assume also negligible re-encryption correctness error and that the level-2 Gaussian error satisfies Equation (33). Then the noise-averaged functional reconstruction attack runs in probabilistic polynomial time. Its guessing advantage in that experiment is greater than $1/4 - \mathsf{negl}(\lambda)$.*

Proof. Use the preprocessing and challenge reconstruction described above with $N_{\mathsf{samp}} = 4d_{\mathsf{rec}}$ and the repetition budget (35). The source and challenge vectors are identically distributed and independent after fixing the public parameters and source key, since neither the message nor the tag enters their first two blocks. The two CRT span events defined above therefore fall under Lemma 3.

Applying Lemma 3 over both fields and taking a union bound gives

$$\Pr[\neg\mathsf{Good}_{\mathsf{span}}] \leq \frac{2d_{\mathsf{rec}}}{N_{\mathsf{samp}} + 1}, \qquad \Pr[\mathsf{Good}_{\mathsf{span}}] > \frac{1}{2},$$

where the second inequality uses $N_{\mathsf{samp}} = 4d_{\mathsf{rec}}$.

By Equation (35) and the polynomial bounds on s_0 and d_{rec}, we have $N_{\mathsf{rep}} = \mathsf{poly}(\lambda)$. For the lifting step, each comparison involves two independent fresh re-encryption-error coordinates $e, e' \leftarrow D_{\mathbb{Z},s_0}$. Equation (34) therefore gives

$$\Pr[\neg\mathsf{Good}_{\mathsf{lift}}] \leq 2N_{\mathsf{samp}}d_{\mathsf{rec}}(N_{\mathsf{rep}} - 1)\varepsilon_{\mathsf{FL}}(\lambda) = \mathsf{negl}(\lambda).$$

For averaging, each error sample is mean-zero subgaussian with parameter s_0. By independence, the average of N_{rep} samples is subgaussian with parameter $s_0/\sqrt{N_{\mathsf{rep}}}$. The standard two-sided subgaussian tail bound therefore gives

$$\Pr\left[\left|\frac{1}{N_{\mathsf{rep}}}\sum_{\nu=1}^{N_{\mathsf{rep}}} e_i^{(\nu)}\right| \geq \frac{1}{2}\right] \leq 2\exp\left(-\frac{\pi N_{\mathsf{rep}}}{4s_0^2}\right).$$

A union bound over all reconstructed coordinates and Equation (35) give

$$\Pr[\neg\mathsf{Good}_{\mathsf{avg}}] \le 2N_{\mathsf{samp}}d_{\mathsf{rec}}\exp\left(-\frac{\pi N_{\mathsf{rep}}}{4s_0^2}\right) \le 2^{-\lambda}.$$

With the same final error, the reconstructed ciphertext is identical to the honest re-encryption of the challenge ciphertext. Since the adversary guesses uniformly on span failure, the above bounds and negligible re-encryption correctness error give guessing advantage at least

$$\frac{1}{2}\Pr[\mathsf{Good}_{\mathsf{span}}] - \mathsf{negl}(\lambda) > \frac{1}{4} - \mathsf{negl}(\lambda).$$

The attack uses $N_{\mathsf{samp}}N_{\mathsf{rep}} = \mathsf{poly}(\lambda)$ re-encryption queries and polynomial-time linear algebra. □

B Partial-Norm Leakage and Full-Block Pseudorandomness

We prove Lemma 4 by conditioning on the public partial norm, applying MS25's arbitrary-error search-to-decision reduction, and preserving the public-key/source split through regular HNF.

B.1 Norm Concentration and Conditional Distributions

Let $(\mathbf{s}, \mathbf{e}_{\mathsf{pk}}) \leftarrow D_{\mathbb{Z}^{2n}, s_{\mathsf{pk}}}$ and $\kappa_{\mathsf{pk}} := \|\mathbf{s}\|_2^2 + \|\mathbf{e}_{\mathsf{pk}}\|_2^2$. Set $\kappa_{\mathsf{max}} := \lfloor 1.28ns_{\mathsf{pk}}^2 \rfloor = \Theta(n^2)$, $\mathcal{K} := \{0, \ldots, \kappa_{\mathsf{max}}\}$, and

$$\mathsf{GoodNorm} := \{\kappa_{\mathsf{pk}} \le \kappa_{\mathsf{max}}\}. \tag{36}$$

MS25 [MS25, Corollary 2] applies to the zero-centred discrete Gaussian for every $s_{\mathsf{pk}} > 0$. Applying it in dimension $2n$ and using that κ_{pk} is an integer gives

$$\Pr[\neg\mathsf{GoodNorm}] = \Pr[\kappa_{\mathsf{pk}} > \kappa_{\mathsf{max}}] \le 2^{-2n}. \tag{37}$$

Moreover, since $s_{\mathsf{pk}} \le \mathrm{poly}(n)$,

$$|\mathcal{K}| = \kappa_{\mathsf{max}} + 1 \le 1.28ns_{\mathsf{pk}}^2 + 1 = \mathrm{poly}(n). \tag{38}$$

The cutoff serves two purposes. For correctness, $\mathsf{GoodNorm}$ implies $s_{\mathsf{ct}}(\mathsf{pk}) \le s_{\mathsf{ct,max}} := \sqrt{2\kappa_{\mathsf{max}}}\, s_{\mathsf{err}} = O(n^{3/2}\log^2 n)$. For security, the unbounded integer Gaussian draws yield a published norm in a polynomial-size set except with negligible probability, providing the polynomial-size range needed for MS25's norm-guessing reduction [MS25, Appendix B.2].

For any polynomially bounded $M \ge 0$, put $m = 2n + M$ and sample $\mathbf{e}_{\mathsf{base}} \leftarrow D_{\mathbb{Z}^M, s_{\mathsf{pk}}}$. Then $\mathbf{x} := (\mathbf{s}, \mathbf{e}_{\mathsf{pk}}, \mathbf{e}_{\mathsf{base}}) \leftarrow D_{\mathbb{Z}^m, s_{\mathsf{pk}}}$ has block lengths n, n, M. For each supported $\kappa \in \mathcal{K}$, let $\mathcal{X}_{\kappa,M}$ denote the conditional distribution on $\mathbb{Z}_q^m$ of $\mathbf{x} \bmod q$ given $\kappa_{\mathsf{pk}} = \kappa$. The conditioning uses the partial norm of the integer lifts before reduction modulo q and may correlate error coordinates.

B.2 MS25 Arbitrary-Error Search-to-Decision

In uniform-secret Gaussian decisional LWE, one distinguishes $(\mathbf{C}, \mathbf{Ct} + \mathbf{x} \bmod q)$ from $(\mathbf{C}, \mathbf{u})$, where $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{m\times n}$, $\mathbf{t} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{x} \leftarrow D_{\mathbb{Z}^m, s_{\mathsf{pk}}}$, and $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m$. MS25 [MS25, Definition 10 and Lemma 9] introduces Known-Norm LWE, which additionally reveals the squared norm of the secret and error. Its small-secret version asks whether

$$(\mathbf{A}, \mathbf{Az} + \mathbf{e} \bmod q, \|\mathbf{z}\|_2^2 + \|\mathbf{e}\|_2^2) \approx_c (\mathbf{A}, \mathbf{u}, \|\mathbf{z}\|_2^2 + \|\mathbf{e}\|_2^2),$$

where $\mathbf{z} \leftarrow \chi^n$, $\mathbf{e} \leftarrow \chi^m$, and $\mathbf{A}$ is uniform. MS25 proves its hardness from small-secret LWE without norm leakage with polynomial loss.

MS25 extends the sample-preserving search-to-decision reduction of Micciancio and Mol [MM11] to arbitrary joint error distributions by observing that the underlying transformations preserve the error vector.

Theorem 8 (Arbitrary-error search-to-decision [MS25, Appendix B, Theorem 6])**.** *Let $q = \mathrm{poly}(n)$ be prime and $m = \mathrm{poly}(n)$ satisfy $m \geq n + \omega(\log n)$. For an arbitrary distribution $\mathcal{X}$ over $\mathbb{Z}_q^m$, sample*

$$\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{m\times n}, \qquad \mathbf{t} \xleftarrow{\$} \mathbb{Z}_q^n, \qquad \mathbf{x} \leftarrow \mathcal{X}, \qquad \mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m.$$

If a PPT algorithm distinguishes

$$(\mathbf{C}, \mathbf{Ct} + \mathbf{x}) \quad \textit{from} \quad (\mathbf{C}, \mathbf{u})$$

with noticeable advantage, then a PPT search algorithm recovers $\mathbf{t}$ from $(\mathbf{C}, \mathbf{Ct} + \mathbf{x})$ with noticeable success probability, for the same $n, m, q, \mathcal{X}$.

B.3 Partial-Norm Leaked Gaussian LWE

Unlike MS25's full error norm, the public leakage $\kappa_{\mathsf{pk}} = \|\mathbf{s}\|_2^2 + \|\mathbf{e}_{\mathsf{pk}}\|_2^2$ excludes the auxiliary source-error block $\mathbf{e}_{\mathsf{base}}$. Sample $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{m\times n}$, $\mathbf{t} \xleftarrow{\$} \mathbb{Z}_q^n$, and $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m$. Here $\mathbf{s}$ is part of the auxiliary error vector $\mathbf{x}$, rather than the uniform LWE secret $\mathbf{t}$. Theorem 8 applies to each fixed conditional distribution $\mathcal{X}_{\kappa,M}$, with $m = 2n + M$.

Corollary 2 (Partial-norm leaked Gaussian LWE)**.** *Assume Gaussian search-LWE with dimension n, prime modulus q, Gaussian width s_{pk}, and polynomially many samples is hard. For any polynomial $M \geq 0$, let $m = 2n + M$ and use the distributions defined above. Then*

$$(\mathbf{C}, \mathbf{Ct} + \mathbf{x} \bmod q, \kappa_{\mathsf{pk}}) \approx_c (\mathbf{C}, \mathbf{u}, \kappa_{\mathsf{pk}}). \tag{39}$$

Proof. For each fixed $\kappa \in \mathcal{K}$, Theorem 8 applies to the conditional error distribution $\mathcal{X}_{\kappa,M}$, including its coordinate correlations. Since $\mathcal{K}$ has polynomial size and its complement has negligible probability, any non-negligible distinguishing advantage would yield a conditional search-LWE solver that succeeds with non-negligible probability on Gaussian search-LWE instances, contradicting the assumption. □

B.4 Regular HNF Transformation

We apply the regular HNF transformation of Applebaum–Cash–Peikert–Sahai [ACPS09, Section 3.1, Lemma 2] to recover the public-key and source LWE blocks from the auxiliary uniform-secret LWE instance. The regular form preserves the coordinate order of the potentially correlated error vector $\mathbf{x} = (\mathbf{s}, \mathbf{e}_{\mathsf{pk}}, \mathbf{e}_{\mathsf{base}})$.

Write $\mathbf{b} = \mathbf{Ct} + \mathbf{x} \bmod q$ and partition the first n rows as

$$\mathbf{C} = \begin{pmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix}, \quad \mathbf{C}_1 \in \mathbb{Z}_q^{n\times n}.$$

If $\mathbf{C}_1$ is invertible, define

$$\mathbf{A}_{\mathsf{H}} := -\mathbf{C}_2\mathbf{C}_1^{-1}, \qquad \mathbf{b}_{\mathsf{H}} := \mathbf{b}_2 + \mathbf{A}_{\mathsf{H}}\mathbf{b}_1 = \mathbf{A}_{\mathsf{H}}\mathbf{s} + \begin{pmatrix} \mathbf{e}_{\mathsf{pk}} \\ \mathbf{e}_{\mathsf{base}} \end{pmatrix} \pmod q.$$

Partitioning the output after its first n coordinates yields

$$\mathbf{A}_{\mathsf{H}} = \begin{pmatrix} \mathbf{A}_0 \\ \mathbf{A}_1 \end{pmatrix}, \qquad \mathbf{b}_{\mathsf{H}} = \begin{pmatrix} \mathbf{A}_0\mathbf{s} + \mathbf{e}_{\mathsf{pk}} \\ \mathbf{A}_1\mathbf{s} + \mathbf{e}_{\mathsf{base}} \end{pmatrix} \pmod q, \tag{40}$$

where $\mathbf{A}_0 \in \mathbb{Z}_q^{n\times n}$ and $\mathbf{A}_1 \in \mathbb{Z}_q^{M\times n}$. Conditional on invertibility, $\mathbf{A}_{\mathsf{H}}$ is uniform and independent of $\mathbf{x}$, while a uniform input $\mathbf{b}$ produces a uniform $\mathbf{b}_{\mathsf{H}}$. The transformation aborts when $\mathbf{C}_1$ is singular. Its nonabort probability is $\prod_{i=1}^{n}(1-q^{-i})$, bounded below by a positive constant. Thus abort causes only a constant-factor reduction loss.

The first n error coordinates become the new secret $\mathbf{s}$; the next n and final M coordinates remain the public-key and source errors, respectively.

B.5 Full-Block Pseudorandomness Proof

We prove Lemma 4 for the joint distributions G_{RR} and G_{UU} defined in (7).

Proof. Apply Corollary 2 followed by the regular HNF transformation above. The real branch gives G_{RR}, while the uniform branch gives G_{UU}, with the supplied norm kept in both. The invertibility event is independent of the Gaussian error vector and its partial norm.

Given a distinguisher for these full-block distributions, the partial-norm reduction tests invertibility, returns an independent uniform bit on abort, and otherwise runs the distinguisher on the transformed tuple. The abort branch has the same output law in both experiments. Hence $G_{\mathsf{RR}} \approx_c G_{\mathsf{UU}}$. □

C Proofs of the LP11 Normal Form and Joint Hiding

We prove the LP11 normal form and combine it with the NEL-LWE reduction recalled in Section 4.4 to establish joint public-key and ciphertext hiding.

C.1 Discrete Gaussian Tools and Conditions

We use the following asymptotic specialisation of the MP13 convolution theorem to the lattice $\mathbb{Z}$, zero shifts, and equal input widths.

Theorem 9 (MP13 scalar convolution [MP13, Theorem 3])**.** *Let $d = d(n) \geq 2$ be polynomially bounded and let $\mathbf{z} \in \mathbb{Z}^d$ be primitive, meaning $\gcd(z_1, \ldots, z_d) = 1$. For a positive negligible function $\epsilon = \epsilon(n)$, suppose*

$$t \geq \sqrt{2}\|\mathbf{z}\|_\infty \eta_\epsilon(\mathbb{Z}).$$

Then, for $\mathbf{x} \leftarrow D_{\mathbb{Z}^d,t}$,

$$\langle \mathbf{z}, \mathbf{x}\rangle \approx_s D_{\mathbb{Z},t\|\mathbf{z}\|_2}.$$

For $\mathbf{w} = (\mathbf{e}_{\mathsf{pk}}, -\mathbf{s})$, set $B_\infty := s_{\mathsf{pk}} \log n$ and define

$$\mathsf{Primitive}(\mathbf{w}) := \{\gcd(w_1, \ldots, w_{2n}) = 1\},$$
$$\mathsf{GoodInf}(\mathbf{w}) := \{\|\mathbf{w}\|_\infty \leq B_\infty\}.$$

Primitivity ensures $\langle \mathbf{w}, \mathbb{Z}^{2n}\rangle = \mathbb{Z}$. Require

$$s_{\mathsf{err}} \geq \sqrt{2}\|\mathbf{w}\|_\infty \eta_\epsilon(\mathbb{Z}). \tag{41}$$

Lemma 1 and a coordinate union bound give $\Pr[\neg\mathsf{GoodInf}] \leq 4ne^{-\pi(\log n)^2} = \mathsf{negl}(n)$. On $\mathsf{GoodInf}$, the right-hand side of (41) is at most $\sqrt{2}B_\infty\eta_\epsilon(\mathbb{Z}) = O(\sqrt{n}\log^2 n)$, so the chosen implicit constant in s_{err} ensures this condition. The same bound applies to $(\mathbf{w}, \mathbf{w})$, which has the same infinity norm. The following lemma shows that primitivity also fails only with negligible probability.

Lemma 8 (Negligible primitive-key failure)**.** *For* $\mathbf{w} \leftarrow D_{\mathbb{Z}^{2n},s_{\mathsf{pk}}}$ *and* $B := ns_{\mathsf{pk}}$*,*

$$\Pr[\neg\mathsf{Primitive}(\mathbf{w})] \le 4ne^{-\pi n^2} + B\left(\frac{1}{2}+\frac{1}{s_{\mathsf{pk}}}\right)^{2n} = 2^{-\Omega(n)}. \tag{42}$$

Proof. By Lemma 1, the event $\|\mathbf{w}\|_\infty > B$ has probability at most $4ne^{-\pi n^2}$. On its complement, every nonprimitive vector has a prime divisor $p \le B$ common to all coordinates, since $B \ge 2$ for sufficiently large n. For each fixed prime p and coordinate i, the event $p \mid w_i$ means that $w_i = kp$ for some $k \in \mathbb{Z}$. By the definition of the discrete Gaussian,

$$\begin{aligned}\Pr[p \mid w_i] &= \sum_{k\in\mathbb{Z}} \Pr[w_i = kp] \\ &= \frac{\sum_{k\in\mathbb{Z}} e^{-\pi(kp)^2/s_{\mathsf{pk}}^2}}{\sum_{z\in\mathbb{Z}} e^{-\pi z^2/s_{\mathsf{pk}}^2}} = \frac{\rho_{s_{\mathsf{pk}}/p}(\mathbb{Z})}{\rho_{s_{\mathsf{pk}}}(\mathbb{Z})}.\end{aligned}$$

For every $t > 0$, Poisson summation [MR07, Lemma 2.8] gives $\rho_t(\mathbb{Z}) = t\rho_{1/t}(\mathbb{Z}) \ge t$, while monotonicity gives $\rho_t(\mathbb{Z}) = 1 + 2\sum_{k\ge1} e^{-\pi k^2/t^2} \le 1 + 2\int_0^\infty e^{-\pi x^2/t^2}\,dx = 1+t$. Since the $2n$ coordinates of $\mathbf{w}$ are independent, multiplying their divisibility probabilities and using $p \ge 2$ gives

$$\Pr[p \mid w_i \text{ for all } i] = \left(\frac{\rho_{s_{\mathsf{pk}}/p}(\mathbb{Z})}{\rho_{s_{\mathsf{pk}}}(\mathbb{Z})}\right)^{2n} \le \left(\frac{1}{p}+\frac{1}{s_{\mathsf{pk}}}\right)^{2n} \le \left(\frac{1}{2}+\frac{1}{s_{\mathsf{pk}}}\right)^{2n}.$$

A union bound over at most B primes yields the second term of (42). Since B is polynomial and $1/2 + 1/s_{\mathsf{pk}} \le 3/4$ eventually, the full bound is exponentially small. □

Taking $M = 0$ in Lemma 4 and discarding the published norm κ_{pk} gives the required small-secret Gaussian decisional LWE premise under the same Gaussian search-LWE assumption. We use the independent-centre sampling of MS25 Definition 11, with an explicit convolution condition for the required width.

Lemma 9 (Fixed-matrix shifted LWE [MS25, Definition 11 and Lemma 10])**.** *Assume small-secret Gaussian decisional LWE with dimension* n*, modulus* q*,* n *scalar samples, and secret and error width* s_{pk}*. Let* $m = \mathrm{poly}(n)$ *and let* Ψ *be an efficiently samplable distribution on* $\mathbb{R}^n$*. Fix a width* $s_{\mathsf{shift}} > s_{\mathsf{pk}}$ *and put* $s_{\mathsf{aux}} := \sqrt{s_{\mathsf{shift}}^2 - s_{\mathsf{pk}}^2}$*. Suppose that, uniformly over the supplied centre* $\mathbf{c}$*, independent summands satisfy*

$$D_{\mathbb{Z}^n,s_{\mathsf{pk}}} + D_{\mathbb{Z}^n,s_{\mathsf{aux}},\mathbf{c}} \approx_s D_{\mathbb{Z}^n,s_{\mathsf{shift}},\mathbf{c}}. \tag{43}$$

Sample one $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times n}$ *and keep it fixed across all columns. Sample all* $\mathbf{c}_i, \mathbf{d}_i \leftarrow \Psi$ *for* $i \in [m]$*. Conditional on these centres, sample*

$$\mathbf{r}_i \leftarrow D_{\mathbb{Z}^n,s_{\mathsf{shift}},\mathbf{c}_i}, \qquad \mathbf{f}_i \leftarrow D_{\mathbb{Z}^n,s_{\mathsf{shift}},\mathbf{d}_i}.$$

Then we have

$$\left(\mathbf{A}, \{(\mathbf{c}_i, \mathbf{d}_i, \mathbf{A}^\top\mathbf{r}_i + \mathbf{f}_i)\}_{i=1}^m\right) \approx_c \left(\mathbf{A}, \{(\mathbf{c}_i, \mathbf{d}_i, \mathbf{u}_i)\}_{i=1}^m\right), \qquad \mathbf{u}_i \xleftarrow{\$} \mathbb{Z}_q^n.$$

Proof. The result follows from the reduction for fixed-matrix shifted LWE of MS25 [MS25, Lemma 10]. We use the auxiliary width s_{aux} specified above. By (43), adding independent shifted Gaussian vectors to the secret and error vectors of LWE instances sharing the same matrix produces the required distributions up to negligible statistical distance. The uniform branch remains uniform, and the standard polynomial-column hybrid preserves computational indistinguishability. □

For the LP11 application, set $s_{\mathsf{shift}} := s_{\mathsf{err}}/\sqrt{2}$ and $s_{\mathsf{aux}} := \sqrt{s_{\mathsf{shift}}^2 - s_{\mathsf{pk}}^2}$. Then $s_{\mathsf{aux}} = \Theta(\sqrt{n}\log^2 n)$ and $(s_{\mathsf{pk}}^{-2} + s_{\mathsf{aux}}^{-2})^{-1/2} = \Theta(\sqrt{n})$. Both exceed $\eta_\epsilon(\mathbb{Z}^n) = O(\log n)$, so Theorem 1, with $s_1 = s_{\mathsf{pk}}$ and $s_2 = s_{\mathsf{aux}}$, gives (43) uniformly over the centre.

C.2 Proof of the LP11 Normal Form with Explicit Noise

We prove Lemma 5 following MS25's normal-form architecture [MS25, Theorem 2], with discrete projection and exact conditioning.

Proof. Let $\mathbf{w} = (\mathbf{e}_{\mathsf{pk}}, -\mathbf{s})$ satisfy the good-key conditions, and write $\mathbf{x} = (\mathbf{r}, \mathbf{f}) \leftarrow D_{\mathbb{Z}^{2n}, s_{\mathsf{err}}}$. The ciphertext noise is $e_{\mathsf{ct}} = \langle \mathbf{w}, \mathbf{x} \rangle + e'$. Sample $\mathbf{h} \leftarrow D_{\mathbb{Z}^{2n}, s_{\mathsf{err}}}$. By Theorem 9, we may replace e' by $\langle \mathbf{w}, \mathbf{h} \rangle$ with negligible statistical distance. Setting $\mathbf{y} = \mathbf{x} + \mathbf{h}$, the resulting noise is $z = \langle \mathbf{w}, \mathbf{y} \rangle$. We make this replacement in both the ciphertext and its auxiliary noise coordinate.

For any fixed $\mathbf{y}$, completing the square gives

$$\mathbf{x} \mid \mathbf{y} \sim D_{\mathbb{Z}^{2n}, s_{\mathsf{err}}/\sqrt{2}, \mathbf{y}/2}. \tag{44}$$

Write $\mathbf{y} = (\mathbf{y}_r, \mathbf{y}_f)$, where $\mathbf{y}_r, \mathbf{y}_f \in \mathbb{Z}^n$ are the blocks corresponding to $\mathbf{r}$ and $\mathbf{f}$, respectively. Conditioned on $\mathbf{y}$, the vectors $\mathbf{r}$ and $\mathbf{f}$ are shifted discrete Gaussians with centres $\mathbf{y}_r/2$ and $\mathbf{y}_f/2$. Their width $s_{\mathsf{shift}} = s_{\mathsf{err}}/\sqrt{2}$ satisfies the convolution conditions verified above. Applying Lemma 9 to all M ciphertexts replaces $\mathbf{a}_i = \mathbf{A}_0^\top \mathbf{r}_i + \mathbf{f}_i$ by independent uniform vectors $\mathbf{u}_i$, while keeping $\mathbf{y}_i$ unchanged. The reduction generates $\mathbf{s}$ and $\mathbf{e}_{\mathsf{pk}}$ itself, so it can compute the public key and each corresponding noise value $z_i = \langle \mathbf{w}, \mathbf{y}_i \rangle$. Each ciphertext/noise pair now has the form

$$((\mathbf{u}_i, \langle \mathbf{u}_i, \mathbf{s} \rangle + \mu_i + z_i) \bmod q, z_i)\,.$$

Finally, since $\mathbf{w}$ is primitive, so is $(\mathbf{w}, \mathbf{w})$. Applying Theorem 9 to the vectors $(\mathbf{x}, \mathbf{h})$ gives

$$z = \langle \mathbf{w}, \mathbf{y} \rangle = \langle (\mathbf{w}, \mathbf{w}), (\mathbf{x}, \mathbf{h}) \rangle \approx_s D_{\mathbb{Z}, \sqrt{2} s_{\mathsf{err}} \|\mathbf{w}\|_2} = D_{\mathbb{Z}, s_{\mathsf{ct}}(\mathsf{pk})}.$$

Since the uniform vectors $\mathbf{u}_i$ are independent of these noise values, we may replace each z_i by an independent $\widetilde{e}_{\mathsf{ct},i} \leftarrow D_{\mathbb{Z}, s_{\mathsf{ct}}(\mathsf{pk})}$, using the same value in the ciphertext and its auxiliary noise coordinate. All statistical errors remain negligible for polynomially many encryptions, and the good-key conditions fail with negligible probability. This proves (8). □

C.3 Proof of Joint Public-Key and Ciphertext Hiding

Proof (Lemma 6). For a positive published norm, take $m = M$, $s_{\mathsf{lwe}} = s_{\mathsf{pk}}$, and $s_{\mathsf{nel}} = s_{\mathsf{ct}}(\mathsf{pk}_B)$ in Theorem 3. With $a_{\mathsf{nel}} = s_{\mathsf{pk}}^2 + 2\eta_\epsilon(\mathbb{Z}^M)^2$, the width conditions (9) hold whenever

$$s_{\mathsf{pk}} \geq \sqrt{2}\eta_\epsilon(\mathbb{Z}^M), \qquad s_{\mathsf{err}}^2 \geq a_{\mathsf{nel}}, \qquad s_{\mathsf{pad}}^2 \geq 2a_{\mathsf{nel}}. \tag{45}$$

Indeed, $s_{\mathsf{ct}}(\mathsf{pk}_B)^2 = 2\kappa_{\mathsf{pk}} s_{\mathsf{err}}^2 \geq 2a_{\mathsf{nel}}$. For our parameters, $\eta_\epsilon(\mathbb{Z}^M)^2 = O(\log^2 n) = o(s_{\mathsf{pk}}^2)$, so $a_{\mathsf{nel}} = (1 + o(1))s_{\mathsf{pk}}^2$. Since $s_{\mathsf{err}}^2 = \Theta(n \log^4 n)$ and $s_{\mathsf{pad}}^2 = 4s_{\mathsf{pk}}^2$, all three inequalities in (45) hold.

Reserve M independent encryptions of zero together with their noise. Lemma 5, followed by adding pads, puts their joint distribution in the form

$$(\mathbf{A}_0, \mathbf{b}_0, \kappa_{\mathsf{pk}}, \mathbf{A}_1, \mathbf{A}_1 \mathbf{s}_B + \widetilde{\mathbf{e}}, \widetilde{\mathbf{e}} + \mathbf{p}),$$

where the rows of $\mathbf{A}_1$ are uniform and $\widetilde{\mathbf{e}} \leftarrow D_{\mathbb{Z}^M, s_{\mathsf{ct}}(\mathsf{pk}_B)}$.

Apply the error-only reduction underlying Theorem 3 to the last vector in Lemma 4, namely $\mathbf{A}_1 \mathbf{s} + \mathbf{e}_{\mathsf{base}}$ in the real experiment and $\mathbf{u}_1$ in the uniform experiment, keeping the other coordinates unchanged. Since the base LWE error $\mathbf{e}_{\mathsf{base}}$ is independent of the key-generation state, the real branch gives the displayed joint distribution up to negligible statistical distance; the uniform branch gives uniform public-key LWE and ciphertext blocks, with the prescribed ideal NEL conditional on the published norm. The claim therefore follows from Lemma 4.

For online payloads, use the exact identity $\mathsf{Enc}_{\mathsf{lp}}(\mathsf{pk}, \mu; \rho) = \mathsf{Enc}_{\mathsf{lp}}(\mathsf{pk}, 0; \rho) + (\mathbf{0}, \mu)$. Fix each payload before revealing its reserved pair, translate the ciphertext, and leave its noise and pad unchanged. □

D Proof of Ordered Multi-Hop HRA Security

Proof (Theorem 5). Use the query bounds and reverse-rank embedding from the proof of Theorem 4. Counting source encryptions, cached rekey components, and fresh encryptions of input pads once per instance gives the per-key and total bounds

$$\begin{aligned} M &= Q_{\mathsf{enc}} + 1 + \ell_{\mathsf{g}}(Q_{\mathsf{users}} - 1) + Q_{\mathsf{reenc}}, \\ N &= Q_{\mathsf{enc}} + 1 + \ell_{\mathsf{g}} Q_{\mathsf{users}}(Q_{\mathsf{users}} - 1) + Q_{\mathsf{reenc}}. \end{aligned} \tag{46}$$

Apply the single-hop pad replacement and public aborts to all three tag namespaces, with tag-collision loss at most $\binom{N}{2}2^{-\lambda}$.

NEL and H2C. Keep the NEL of every instance and define the NEL of derived records by (27), preserving values across reused ancestors and cached components. Proposition 1 extends to admissible derived inputs as

$$c_C = (\mathbf{0}, \mu_A + \ell_{\mathsf{ct},A}) - \sum_j d_{A,j} K^{(0)}_{AC,j} + P^{(0)}_{AC,k} \pmod q. \tag{47}$$

Keep the same coins, pads, and output header, with cached $K^{(0)}_{AC,j}$ and fresh $P^{(0)}_{AC,k}$. In the real scheme, each record's pad is fixed by its header and the PRF key at creation and computed when the record is re-encrypted. For corrupted and unprocessed honest owners, the phase invariant $\phi_{\mathbf{s}_A}(c_A) \equiv \mu_A + \ell_{\mathsf{ct},A} - p_{\mathsf{ct},A} \pmod q$ propagates through real switching and this rewrite, since

$$\phi_{\mathbf{s}_C}(c_C) \equiv \mu_A + \ell_{\mathsf{ct},A} - \sum_j d_{A,j} e_{\mathsf{rk},j} + e_{\mathsf{fr}} \equiv \mu_A + \ell_{\mathsf{ct},C} - p_{\mathsf{ct},C} \pmod q.$$

For processed owners, Equation (47) defines H2C replies using record NEL.

Online rank step. Fix β and write G^β_r for the visible transcript and final guess after processing ranks above r. Lemma 6, applied to the current key and its at most M instances, gives

$$G^\beta_r \approx_c G^\beta_{r-1}. \tag{48}$$

At a processed target, output record k has

$$c_B = -\sum_j d_{A,j} U_{AB,j} + V_{AB,k}, \tag{49}$$

with cached $U_{AB,j}$ and fresh uniform $V_{AB,k}$ conditional on earlier data, published norms, and the NEL of all activated instances. The public shift $(\mathbf{0}, v_A)$ is absorbed into $V_{AB,k}$.

Lower-ranked and corrupted sources have available secrets and pads; Equation (22) also recovers derived pads from instances under their current owner. Thus the simulator computes each incoming payload from the interaction so far, before seeing the ciphertext and NEL of the reserved instance used to encrypt it. Outgoing H2H and H2C use (49) and (47) without the current secret or separate pads. The phase invariant covers returns through corrupted users.

Final game. Honest source LP11 ciphertexts and the fresh residuals at honest targets are uniform. The NEL recurrence uses only the NEL kept in the proof and public decompositions, while H2C replies use known nonchallenge messages. Query induction therefore extends through later hops and branching, giving

$$G_0^0 \equiv G_0^1 \tag{50}$$

for the complete visible transcripts and final guesses. □

E Proof of the Key-Separation Compiler Theorem

Proof (Theorem 6). Fix a PPT adversary $\mathcal{A}$ against the compiled scheme, with registration bounds $q_{\mathsf{H}}, q_{\mathsf{C}}$. We construct an adversary $\mathcal{B}$ for the ordered base experiment. Before answering any registration query from $\mathcal{A}$, $\mathcal{B}$ registers all base users in increasing order of level: first q_{H} honest and q_{C} corrupted users for level 0, then the same numbers for level 1, and so on through level t. In particular, every level-h base user is registered before every level-$(h+1)$ base user. To register an honest user, $\mathcal{B}$ selects one unused honest base user from each level and returns their public keys. To register a corrupted user, it selects one unused corrupted base user from each level and returns both their public and secret keys. Since the keys are independent, this gives the correct registration distribution.

The reduction forwards encryption and challenge queries for user A to $\mathsf{pk}_{A,0}$. A permitted rekey release for $A \to B$ returns the t base rekeys from $\mathsf{pk}_{A,h}$ to $\mathsf{pk}_{B,h+1}$, for $0 \le h < t$. Re-encryption of a depth-h record uses the corresponding base edge. Each such H2H edge goes from the base user assigned to A at level h to the base user assigned to B at level $h+1$. By the registration order above, its source was registered before its target, as required by the ordered base experiment.

Each record in the compiled experiment under A at depth h corresponds to a distinct base record under $\mathsf{pk}_{A,h}$, with the same ciphertext, depth, and challenge-descendant status. The reduction rejects invalid queries in the compiled experiment locally, including self-delegations. The record correspondence and unchanged corruption classes preserve the challenge-derived H2C prohibition and all other query restrictions.

The base challenger generates and caches the required rekey components. All re-encryptions use the same base proxy state, so its updates on success or failure are also preserved. Thus, for either challenge bit, $\mathcal{A}$ sees exactly the compiled experiment. Returning its guess gives $\mathsf{Adv}^{\mathsf{hra},t}_{\Pi^{t\text{-hop}}}(\mathcal{A}) = \mathsf{Adv}^{\mathsf{ord\text{-}hra},t}_{\Pi}(\mathcal{B})$. The registration bounds are multiplied by $t+1$ and each rekey release uses at most t base queries; other query counts are unchanged. Hence $\mathcal{B}$ is PPT for polynomial t. □