Post-Quantum Multi-Message Public Key Encryption from Extended Reproducible PKE

Hongxiao Wang[1], Ron Steinfeld[2], Markku-Juhani O. Saarinen[3], Muhammed F. Esgin[2], and Siu-Ming Yiu[1(✉)]

[1] The University of Hong Kong, China
{hxwang, smyiu}@cs.hku.hk
[2] Monash University, Australia
{Ron.Steinfeld, Muhammed.Esgin}@monash.edu
[3] Tampere University, Finland
markku-juhani.saarinen@tuni.fi

Abstract. A multi-message multi-recipient Public Key Encryption (mmPKE) enables batch encryption of multiple messages for multiple independent recipients in one go, significantly reducing costs, particularly bandwidth, compared to the trivial solution of encrypting each message individually. This capability is especially critical in the post-quantum setting, where ciphertext length is typically significantly larger than the corresponding plaintext.

In this work, we first observe that the generic construction of mmPKE from *reproducible PKE* proposed by Bellare et al. (PKC '03) does not apply in the lattice-based setting because existing lattice-based PKE schemes do not fit the notion of reproducible PKE. To this end, we first extend their construction by proposing a new variant of PKE, named *extended reproducible PKE (XR-PKE)*, which enables the reproduction of ciphertexts via additional hints. However, standard lattice-based PKE schemes, such as Kyber (EuroS&P '18), do not readily satisfy the XR-PKE requirements. To construct XR-PKE from lattices, we introduce a novel technique for precisely estimating the impact of such hints on the ciphertext security while also establishing suitable parameters. This enables us to instantiate the *first* CPA-secure mmPKE and Multi-Key Encapsulation Mechanism (mmKEM) from the *standard* Module Learning with Errors (MLWE) lattice assumption, named mmCipher-PKE and mmCipher-KEM, respectively. We then extend our works to the identity-based setting and construct the *first* mmIBE and mmIB-KEM schemes. As a bonus contribution, we explore generic constructions of *adaptively secure* mmPKE, achieving security against adaptive corruption and chosen-ciphertext attacks.

We also provide an efficient implementation and thorough evaluation of the practical performance of our mmCipher. Our results show that mmCipher provides significant bandwidth and computational savings in practice, compared to the state-of-the-art. For example, for 1024 recipients, our mmCipher-KEM achieves a 23–45× reduction in bandwidth overhead, reaching within 4–9% of the plaintext length (*near optimal bandwidth*), while also offering a 3–5× reduction in computational cost.

Keywords: Public Key Encryption · mmPKE · Post-Quantum · Lattice

1 Introduction

Public Key Encryption (PKE) and Key Encapsulation Mechanism (KEM) are foundational cryptographic primitives essential for secure digital communication. The rapid progress in quantum computing [23] has led to a shift towards post-quantum cryptography. In response, the National Institute of Standards and Technology (NIST) has selected Kyber, a lattice-based KEM/PKE, as a primary candidate for standardization [2]. However, these quantum-resistant constructions generally require significantly more bandwidth resources than their traditional counterparts [8]. Therefore, reducing communication costs for multiple recipients, even for moderately large number of recipients, say $N \geqslant 16$, is already of practical significance.

Multi-message multi-recipient PKE. To address this need, multi-message multi-recipient PKE (mmPKE) was introduced to efficiently batch multiple ciphertexts when sending distinct messages to multiple recipients by Kurosawa [43]. Unlike trivial solutions where each message is encrypted separately, mmPKE allows for significant bandwidth savings, especially valuable in post-quantum settings where ciphertext sizes are large. We call an mmPKE (asymptotically) *bandwidth-optimal* if the length of its ciphertexts approaches the total length of its plaintexts (for a large number of recipients). When each message is an encapsulated key, we obtain the multi-key multi-recipient KEM (mmKEM). A special case of mmPKE and mmKEM is multi-recipient PKE (mPKE) and multi-recipient KEM (mKEM), which only support sending the same message or encapsulated key to all recipients. In this case, since each recipient receives the same message, the security model excludes the insider adversaries (recipients).

Due to its practically appealing and theoretically interesting nature, the studies of mmPKE/mmKEM [5,9,10,43,60], mPKE/mKEM [7,19,35,39,48,62,65], and their identity-based variants [18, 36, 55] have attracted significant attention. Building on this, (m)mPKE plays a central role in Secure Group Messaging (SGM) [4, 35, 39] and confidential transactions [16, 24, 34] [4]. Among them, the foundational work on mmPKE was proposed by Bellare et al. in [9, 10] that significantly expanded Kurosawa's work by: (1) introducing the *insider adversary* to formalize the mmPKE security model, (2) identifying possible attacks (e.g., rogue public key attacks) and introducing the *knowledge-of-secret-key* (KOSK) assumption (i.e., the challenger knowing the private key of each public key) for protection, and (3) defining the *reproducible PKE* to generically construct mmPKE. Then, they show that many popular discrete-log-based encryption schemes, such as ElGamal [26] and Cramer-Shoup [21], are reproducible and can be extended to mmPKE under the KOSK assumption. Informally, reproducibility requires that there exists an efficient algorithm to transform a ciphertext into another ciphertext for a different public key and message while using the same randomness. Unfortunately, such properties are *not known* to exist for post-

[4] These confidential transactions implicitly employ mmPKE, i.e., they directly utilize ElGamal-based mmPKE as a fundamental building block.

quantum assumptions, particularly for lattice-based assumptions, since some fresh error/noise in each ciphertext is necessary and cannot be fully eliminated.

Currently, the only known post-quantum mmPKE [5] is generically constructed from mKEM, but it *only supports* batching *consecutive identical* messages in the message vector. Here, we identify two key limitations of this approach: (1) its efficiency is *close to trivial solution* when messages are independent, and (2) it *cannot achieve full CPA* security, as it leaks the structure of the input message vector, i.e., given the multi-recipient ciphertext, others can identify whether any two consecutive messages in the message vector are identical. This significantly limits the application of [5] in many practical scenarios, such as confidential transactions [16, 24, 34]. For example, the anonymity of spender and recipient in anonymous payment systems like anonymous Zether [24] crucially relies on the *full* CPA security of ElGamal-based mmPKE (where identical encrypted zero messages correspond to 'decoy' accounts used to protect anonymity of the spender and recipient accounts).

Thus, despite recent progress, significant challenges remain in fully realizing the potential of mmPKE in post-quantum settings, especially for generic constructions, leading to our question:

Question: Are there any simple and efficient generic constructions of fully batched mmPKE based on the post-quantum assumptions, while enjoying full CPA-security, regardless of the message vector?

The above question also applies to post-quantum mmIBE schemes.

We refer to the first rows of Table 1 and Table 2 for a summary of the large ciphertext expansion rate and computation costs of the existing trivial post-quantum mmPKE solutions. We note that this comparison excludes [5], as their benchmarks only focus on mKEM, which we consider incomparable to the case of mmKEM/mmPKE. Furthermore, in our setting, since the messages/keys are *independent* of each other, the probability of consecutive identical messages appearing in the message vector is *negligible*. Therefore, as [5] only supports batching consecutive identical messages, its performance under independent messages would be equivalent to the trivial solution with Kyber.

1.1 Contribution

In this work, we revisit the mmPKE paradigm built from reproducible PKE [10] and show how to extend it to the post-quantum setting via a new variant of PKE, called *extended reproducible PKE (XR-PKE)*. We then construct a family of XR-PKE schemes from lattices and give efficient instantiations of mmPKE based on *standard* MLWE assumption, along with the identity-based variants. Lastly, we provide an efficient implementation and its performance evaluation for our mmPKE. We summarize the main contributions of our work in the following. For further technical details of our contributions, we refer to Section 1.2.

Generic construction of post-quantum mmPKE. Our first contribution is a generic construction of post-quantum mmPKE from XR-PKE. To this end, we formally define the notion of XR-PKE that significantly enhances the functionality

Table 1: Comparison of current lattice-based CPA-secure mmPKE/mmKEM schemes, for $N = 1024$ recipients.

Scheme	PQ-Sec. Level	Enc. Size (KB)	Enc. Exp. Factor	Improve (×)	Enc. Time (ms)	Full CPA
*Plaintext**	–	32	1.0	–	–	✗
Baseline:	128	768	24.0	–	36	
Kyber [14]	192	1088	34.0	–	58	✓
(ML-KEM [54])	256	1568	49.0	–	87	
Our work:	128	**33**	**1.0**	**23.1×**	12	
mmCipher-KEM	192	**34**	**1.1**	**31.6×**	16	✓
(Cons. B.2+4.8)	256	**35**	**1.1**	**44.7×**	17	
Our work:	128	65	2.0	11.8×	13	
mmCipher-PKE	192	66	2.1	16.4×	16	✓
(Cons. 3.4+4.4)	256	67	2.1	23.3×	18	

* Here, Enc. Size represents the size of all encapsulated keys/messages in plaintext.

Notes: For each scheme, we report the size of the multi-recipient ciphertext (Enc. Size) in kilobytes (KB) as well as the improvement in the ciphertext expansion factor (Enc. Exp. Factor), relative to the trivial solution with CPA-secure Kyber (parameterized by ML-KEM standard [54]), and the encryption/encapsulation time (Enc. Time) in milliseconds (ms), under 128-bit, 192-bit, and 256-bit post-quantum security levels (PQ-Sec. Level), respectively. Each message/key is 256 bits and *independently* chosen across 1024 recipients. *Full CPA* indicates that the scheme protects both semantics and structure of the message vector.

of the original reproducible PKE [10], both in syntax (by incorporating hints into the reproduction algorithm and providing a hint generation algorithm) and in security model (by modeling the semantic security of ciphertexts given the associated hints). Furthermore, we extend our results to mmKEM and mmIBE/mmIB-KEM settings. We believe that such a new variant of PKE may be of independent interest for other cryptographic constructions.

mmCipher: efficient mmPKE instantiations from lattices. Our second contribution is the construction of lattice-based XR-PKE and XR-KEM schemes, from which we instantiate mmPKE based on lattices. At a high level, to achieve extended reproducibility, we use the decryption error as the hint to assist in reproducing the ciphertext. To establish the semantic security of ciphertexts given the associated hints, we rely on the Matrix Hint-MLWE assumption [29], for which there exists a reduction from the standard MLWE assumption (under appropriate parameter choices). This allows us to argue that the influence of the hints on the semantic security of ciphertext is negligible. Along the way, as a small bonus technical contribution, we also identify a missing efficient sampleability condition in the parameter instantiation for the reduction of [29]. The sampleability condition on the refined Matrix Hint-MLWE security reduction may be of independent interest in other applications of Hint-MLWE, e.g., [1,42,45].

Table 2: Comparison of current lattice-based CPA-secure mmIBE/mmIB-KEM schemes, for $N = 1024$ recipients.

Scheme	PQ-Sec. Level	Message Space	Enc. Size (KB)	Enc. Exp. Factor	Improve (×)	Full CPA
Baseline:	80	$\{0,1\}^{512}$	1664	26.0	–	✓
DLP IBE [25]	192	$\{0,1\}^{1024}$	3840	30.0	–	
Our work:	80	$\{0,1\}^{1024}$	**137**	**1.1**	**24.1**×	✓
mmCipher-KEM with [20]	192	$\{0,1\}^{2048}$	**274**	**1.1**	**27.3**×	
Our work:	80	$\{0,1\}^{1024}$	265	2.1	12.5×	✓
mmCipher-PKE with [20]	192	$\{0,1\}^{2048}$	530	2.1	14.3×	

Notes: For each scheme, we report the size of the multi-recipient ciphertext (Enc. Size) in kilobytes (KB), as well as the message space. To provide a fair comparison across different message spaces, we report the improvement in the ciphertext expansion factor (Enc. Exp. Factor), relative to the trivial solution with CPA-secure DLP IBE [25], under 80-bit and 192-bit post-quantum security levels (PQ-Sec. Level), respectively. Each message/key is *independently* chosen across 1024 recipients. *Full CPA* indicates that the scheme protects both semantics and structure of the message vector.

Then, following our generic construction, we instantiate two lattice-based CPA-secure mmPKE under the KOSK assumption: (1) an mmPKE for short messages (mmCipher-PKE) and (2) a hybrid mmKEM-DEM scheme for arbitrary-length messages (mmCipher-KEM), enjoying the following features:

- *Enhanced Security:* Both our mmCipher-KEM and mmCipher-PKE achieve *full CPA-security* which can protect both *semantics* and *structure* of the input message vector, preventing the identification of identical messages. This is a key improvement over [5], broadening potential applications.
- *IBE Compatibility:* Both mmCipher-KEM and mmCipher-PKE are compatible with the identity-based setting. As noted in Remark 4.9, by leveraging the standard pre-image sampling algorithm [20], we develop lattice-based mmIBE and mmIB-KEM. These constructions achieve bandwidth overhead reductions of 13–14× and 24–27×, respectively, for $N = 1024$ recipients at different security levels (80- and 192-bit), compared to the trivial solution with DLP IBE [25].

Additionally, when our mmCipher-PKE functions as an mmKEM, it enables flexible key encapsulation for recipients, such as sending the same key to some recipients without revealing their identities.

Furthermore, to fit the real-world applications, we introduce a compiler that removes the KOSK assumption from mmPKE/mmKEM with *polynomial-size* number of recipients by leveraging a *multi-proof extractable* Non-Interactive Zero-Knowledge (NIZK) proof system. Note that while [10] observed that the KOSK assumption could be removed using NIZK, no concrete construction or formal security proof was given prior to this work.

Practical mmPKE implementation and evaluation. We also provide a C implementation[5] of our lattice-based mmPKE schemes (i.e., mmCipher), together with computational performance and bandwidth benchmarks. Compared to the state-of-the-art, the performance of our mmCipher is independent of the structure of the message vector, i.e., whether the message vector has identical or distinct messages. For $N = 1024$ recipients and different security levels (128-, 192-, 256-bit), our mmCipher-KEM and mmCipher-PKE achieve a 23–45× and 12–23× reduction in bandwidth overhead, respectively, and offer a 3–5× reduction in computational cost, compared to [5] with independent messages and the trivial solution with Kyber. Notably, using a reconciliation mechanism [57], each *public-key-dependent* ciphertext in our mmCipher-KEM is minimized to the size of the encapsulated key (e.g., 256 bits), making our construction *asymptotically-bandwidth-optimal*, with ciphertext size within 4% (resp. 9%) of the plaintext size for 128-bit (resp. 256-bit) security levels and $N = 1024$ recipients.

Generic construction of adaptively secure mmPKE. As a bonus contribution, we propose generic constructions that transform the CPA-secure mmPKE into an adaptively secure mmPKE, achieving security against adaptive corruption and CCA. Specifically, due to the absence of fully batched post-quantum mmPKE constructions, there remains a gap in achieving *adaptive* security in such settings. For example, since the public parameters and randomness are shared among recipients, standard techniques such as the Fujisaki-Okamoto (FO) transform [32, 64], lossy trapdoor functions [58, 63], and the BCHK transform via IBE [12, 17, 49] cannot be applied in the post-quantum mmPKE setting. To this end, we generalize the Naor-Yung paradigm [51, 61] to the mmPKE setting. Furthermore, by leveraging the structure of mmPKE, we can *safely merge* the two ciphertexts into a *single* multi-recipient ciphertext by doubling recipient number from N to $2N$. As a result, only one public-key-independent ciphertext needs to be generated, significantly reducing overhead. The detailed construction is provided in Appendix D.

1.2 Technical Overview

We begin by recalling the syntax of mmPKE [10]. Informally, the setup, key generation and decryption algorithms of mmPKE are the same as the ones in the standard PKE. For the multi-encryption, i.e., $\mathbf{ct} \leftarrow \mathsf{mmEnc}(\mathsf{pp}, (\mathsf{pk}_i)_{i\in[N]}, (\mathsf{m}_i)_{i\in[N]})$, it takes as input the public parameter pp, a set of public keys $(\mathsf{pk}_i)_{i\in[N]}$ along with a set of corresponding messages $(\mathsf{m}_i)_{i\in[N]}$ and outputs a multi-recipient ciphertext $\mathbf{ct}$. The multi-recipient ciphertext $\mathbf{ct}$ can later be extracted to the individual ciphertext ct_i for the public key pk_i by some extraction algorithm.

The correctness of mmPKE is that each individual ciphertext ct_i can be successfully decrypted to the message m_i by the corresponding private key sk_i.

The IND-CPA (under KOSK assumption) security model of of mmPKE is more complicated than standard PKE, since it considers the insider attack where the adversary is allowed to generated some public keys for the challenger to

[5] Our implementation: https://github.com/ml-kem/mmcipher-artifact

encrypt the challenge ciphertext. Informally, the adversary selects ℓ honestly generated (challenger's) public keys $(\mathsf{pk}_i)_{i\in[\ell]}$ and ℓ message pairs $(\mathsf{m}_i^0, \mathsf{m}_i^1)_{i\in[\ell]}$ and $N-\ell$ adversarially generated (adversary's) public keys $(\mathsf{pk}_i)_{i\in[\ell:N]}$ (along with the corresponding private keys $(\mathsf{sk}_i)_{i\in[\ell:N]}$) and messages $(\mathsf{m}_i)_{i\in[\ell:N]}$, and it should be infeasible for the adversary to distinguish the challenge ciphertext $\mathbf{ct} = \mathsf{mmEnc}(\mathsf{pp}, (\mathsf{pk}_i)_{i\in[N]}, (\mathsf{m}_i^b)_{i\in[\ell]}, (\mathsf{m}_i)_{i\in[\ell:N]})$ for a randomly chosen bit $b \in \{0,1\}$.

Traditional mmPKE from reproducible PKE. We recall the traditional constructions of mmPKE from reproducible PKE [10]. The syntax, correctness and security definition of reproducible PKE is the same as standard PKE, except introducing a reproducibility property.

The reproducibility requires that given a ciphertext $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pp}, \mathsf{pk}, \mathsf{m}; \mathsf{r})$ which encrypts the message m with the public key pk and some randomness r, there exists a PPT algorithm, called reproduction algorithm, satisfying

$$\mathsf{Enc}(\mathsf{pp}, \mathsf{pk}', \mathsf{m}'; \mathsf{r}) = \mathsf{Rep}(\mathsf{pp}, \mathsf{ct}, \mathsf{m}', \mathsf{sk}', \mathsf{pk}').$$

It means that the Rep algorithm can use the private key sk' to reproduce a ciphertext ct to another ciphertext ct' for the corresponding public key pk' and different message m' but with the same randomness r. For example, for the ElGamal scheme, given a ciphertext $(g^r, m \cdot (g^x)^r)$ for public key g^x and message m, the other ciphertext for public key $g^{x'}$ and message m' can be reproduced as $(g^r, m' \cdot (g^r)^{x'})$ by the private key x'.

Now, let us discuss how [10] constructs an mmPKE from reproducible PKE. The setup, key generation, and decryption algorithms of mmPKE are the same as the ones in reproducible PKE. In multi-encryption, it uses the same randomness r to encrypt each message m_i for the corresponding public key pk_i to the ciphertext $\mathsf{ct}_i \leftarrow \mathsf{Enc}(\mathsf{pp}, \mathsf{pk}_i, \mathsf{m}_i; \mathsf{r})$ and concatenate the ciphertexts together as multi-recipient ciphertext $\mathbf{ct} := (\mathsf{ct}_1, \ldots, \mathsf{ct}_N)$.

Note that if all ct_i have a same part due to the randomness reuse, this part only needs to be computed and communicated once in the multi-recipient ciphertext and that is the reason for the bandwidth and computation savings of the mmPKE. For example, the part g^r of the ciphertext only needs to be generated once in ElGamal-based mmPKE which can save about half bandwidth and computation compared to the trivial solution.

To reduce the security of mmPKE to that of the underlying reproducible PKE, the reduction follows the same structure as mmPKE, except that it generates the multi-recipient ciphertext by reproducing its own challenge ciphertext for the other recipients. For details, we refer the reader to [10, Theorem 6.2].

Post-quantum mmPKE from XR-PKE. The major limitation of the above mmPKE [10] is that it does not seem to extend to the post-quantum setting, especially lattice-based setting. The reason is that the randomness of the ciphertext in lattice-based PKE schemes cannot be fully reused as in the classical Diffie-Hellman type assumptions. In particular, in encryption scheme based on the LWE lattice problem, the ciphertext for message m typically takes the form $(\mathbf{Ar} + \mathbf{e}_u, \langle \mathbf{b}, \mathbf{r} \rangle + y + \lfloor q/2 \rfloor \cdot m)$. For security, the message error term y cannot

be reused across multiple messages/recipient public keys. Moreover, there are additional reproducibility security issues caused by such error terms.

To get around this issue, we first consider the (extended) reproducible PKE in a *decomposable* variant. Informally, a decomposable encryption algorithm Enc takes as input the randomness $\mathsf{r} := (\mathsf{r}_0, \hat{\mathsf{r}})$ and creates a public-key-*independent* ciphertext $\mathsf{ct}_0 \leftarrow \mathsf{Enc}^{\mathsf{i}}(\mathsf{pp}; \mathsf{r}_0)$ and public-key-*dependent* ciphertext $\widehat{\mathsf{ct}} \leftarrow \mathsf{Enc}^{\mathsf{d}}(\mathsf{pp}, \mathsf{pk}, \mathsf{m}; \mathsf{r}_0, \hat{\mathsf{r}})$. Note that the randomness $\hat{\mathsf{r}}$ in key-dependent ciphertext can be set empty, i.e., $\hat{\mathsf{r}} := \perp$, if it is unnecessary. We view this as a natural formalization of (extended) reproducible PKE as it is satisfied by all the constructions that we are aware of.

Therefore, we intend to reuse only the randomness r_0 in key-independent ciphertext instead of the entire randomness $\mathsf{r} = (\mathsf{r}_0, \hat{\mathsf{r}})$, so that we can achieve the same savings in bandwidth and computation as fully reusing the randomness when constructing mmPKE. We formalize this new primitive, called XR-PKE, which significantly improves upon reproducible PKE in both syntax and security model.

From the perspective of syntax, to formalize the property of reproducibility, we introduce an additional input h', called *hint*, into the reproduction algorithm. Looking ahead to our lattice-based instantiation, the hint there will be used to provide randomized information on the ciphertext error terms needed to reproduce the ciphertext for new recipient. We require that, given a ciphertext $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pp}, \mathsf{pk}, \mathsf{m}; \mathsf{r}_0, \hat{\mathsf{r}})$, the following property always holds

$$\mathsf{Enc}(\mathsf{pp}, \mathsf{pk}', \mathsf{m}'; \mathsf{r}_0, \hat{\mathsf{r}}') = \mathsf{Rep}(\mathsf{pp}, \mathsf{ct}, \mathsf{m}', \mathsf{pk}', \mathsf{sk}', \mathsf{h}').$$

Additionally, we provide an auxiliary algorithm, named *hint generation* algorithm, for generating the hint h'. It takes as input the public parameter pp, the reused randomness r_0, a fresh randomness $\hat{\mathsf{r}}'$, and a public-private key pair $(\mathsf{pk}', \mathsf{sk}')$, i.e.,

$$\mathsf{h}' \leftarrow \mathsf{HintGen}(\mathsf{pp}, \mathsf{r}_0, \hat{\mathsf{r}}', \mathsf{pk}', \mathsf{sk}').$$

Regarding the security model, we require that the adversary's advantage against semantic security remains negligible, even given the hints associated with the challenge ciphertext. More precisely, we introduce a *hint query phase* before the adversary output phase in the security game. In the hint query phase, after receiving the challenge ciphertext $\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{pp}, \mathsf{pk}^*, \mathsf{m}_b^*; \mathsf{r}_0, \hat{\mathsf{r}}^*)$, the adversary is allowed to query N hints on the challenge ciphertext by N public-private key pairs $(\mathsf{pk}_i, \mathsf{sk}_i)_{i\in[N]}$. The challenger then computes the hints as $(\mathsf{h}_i)_{i\in[N]} \leftarrow \mathsf{HintGen}(\mathsf{pp}, \mathsf{r}_0, (\hat{\mathsf{r}}_i)_{i\in[N]}, (\mathsf{pk}_i, \mathsf{sk}_i)_{i\in[N]})$ and returns them to the adversary. The formal definitions of XR-PKE are provided in Section 3.

We now describe the generic construction of post-quantum mmPKE from XR-PKE. The setup, key generation, and decryption algorithms are identical to those in XR-PKE, except that the setup algorithm additionally takes the recipient number N as input. In the multi-encryption algorithm mmEnc, the randomness is structured as $\mathsf{r} = (\mathsf{r}_0, \hat{\mathsf{r}}_1, \ldots, \hat{\mathsf{r}}_N)$. The algorithm first generates an key-independent ciphertext $\mathsf{ct}_0 \leftarrow \mathsf{Enc}^{\mathsf{i}}(\mathsf{pp}; \mathsf{r}_0)$, then computes N key-dependent ciphertexts $\widehat{\mathsf{ct}}_i \leftarrow \mathsf{Enc}^{\mathsf{d}}(\mathsf{pp}, \mathsf{pk}_i, \mathsf{m}_i; \mathsf{r}_0, \hat{\mathsf{r}}_i)$, and concatenates them as multi-recipient ciphertext $\mathbf{ct} :=$

$(\mathsf{ct}_0, \widehat{\mathsf{ct}}_1, \ldots, \widehat{\mathsf{ct}}_N)$. For each recipient, the individual ciphertext $\mathsf{ct}_i := (\mathsf{ct}_0, \widehat{\mathsf{ct}}_i)$ can be extracted from $\mathbf{ct}$ and decrypted by the private key sk_i.

Finally, we briefly discuss the security reduction from mmPKE to the underlying XR-PKE. The reduction follows the same structure as that in traditional mmPKE built from reproducible PKE [10], except that, prior to reproducing the ciphertext, it must send the public-private key pairs $(\mathsf{pk}_i, \mathsf{sk}_i)_{i\in[N]}$ to its challenger during the hint query phase, and receive the corresponding hints $(\mathsf{h}_i)_{i\in[N]}$ in order to complete the reproduction. We provide the detailed proof in Theorem 3.5. We note that our mmPKE construction as described here relies on the KOSK assumption, and we show how to explicitly remove this requirement via the NIZK in Section 4.4.

Constructing lattice-based XR-PKE. To the best of our knowledge, no existing lattice-based PKE schemes currently satisfy the extended reproducibility property. The primary reason is that they fail to achieve semantic security of the ciphertext given the associated hints. In the following, we present how to construct a lattice-based XR-PKE under the standard MLWE assumption at a high level. We believe that our approach may be of independent interest, as it applies to both plain and ring-based lattice settings.

We begin with one of the most efficient lattice-based PKE, the finalist in the NIST post-quantum cryptography competition, Kyber [14] as follows, and show how to transform it into XR-PKE.

At the beginning, a uniformly random matrix $\mathbf{A} \leftarrow \mathcal{U}(\mathcal{R}_q^{m\times n})$ is sampled as the public parameter where $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^d+1)$ and $\mathcal{R} = \mathbb{Z}[X]/(X^d+1)$. Then, the public key is generated by

$$\mathbf{b} := \mathbf{A}^\top \mathbf{s} + \mathbf{e} \tag{1.1}$$

where the private key $(\mathbf{s}, \mathbf{e}) \leftarrow \mathcal{U}(\mathbb{S}_\nu^m) \times \mathcal{U}(\mathbb{S}_\nu^n)$ has coefficients uniformly randomly sampled from set $[-\nu, ..., \nu]$ for $\nu \ll q$. To encrypt a message m, the ciphertext can be *decomposed* into two parts: a key-*independent* ciphertext $\mathbf{c}$, a key-*dependent* ciphertexts u as below,

$$\mathbf{c} := \mathbf{A}\mathbf{r} + \mathbf{e}_u, \qquad u := \langle \mathbf{b}, \mathbf{r}\rangle + y + \lfloor q/2 \rceil \cdot m, \tag{1.2}$$

where randomness are sampled from some distribution χ over $\mathcal{R}$ as $\mathbf{r} \leftarrow \chi^n$, $\mathbf{e}_u \leftarrow \chi^m$, $y \leftarrow \chi$, and $m \in \{0,1\}^d$ (interpreted as a polynomial in $\mathcal{R}$ with binary coefficients). To decrypt the ciphertext $(\mathbf{c}, u)$ to the message m, the recipient uses the private key to compute the $u - \langle \mathbf{c}, \mathbf{s}\rangle$. Using Equations (1.1) and (1.2), we have

$$u - \langle \mathbf{c}, \mathbf{s}\rangle = \langle -\mathbf{s} || \mathbf{e}, \mathbf{e}_u || \mathbf{r}\rangle + y + \lfloor q/2 \rceil \cdot m.$$

where $||$ denotes the usual concatenation. If the PKE is correct, i.e., $\|\langle -\mathbf{s}||\mathbf{e}, \mathbf{e}_u||\mathbf{r}\rangle + y\|_\infty \leqslant \lfloor q/4 \rceil$, after rounding the above term as $\lfloor u - \langle \mathbf{c}, \mathbf{s}\rangle \rceil_2$, each recipient can obtain the message m and the decryption error $h := \langle -\mathbf{s}||\mathbf{e}, \mathbf{e}_u||\mathbf{r}\rangle + y$.

Here we use the decryption error h as the hint to reproduce the ciphertext. Given a ciphertext $(\mathbf{c}, u)$, a new ciphertext $(\mathbf{c}, u')$ for another public key $\mathbf{b}' =$

$\mathbf{A}^\top \mathbf{s}' + \mathbf{e}'$ and message m' using randomness $((\mathbf{r}, \mathbf{e}_u), y')$ can be reproduced by the corresponding private key $\mathbf{s}'$ and the hint h' as

$$u' := \langle \mathbf{c}, \mathbf{s}' \rangle + h' + \lfloor q/2 \rceil \cdot m' = \langle \mathbf{b}', \mathbf{r} \rangle + y' + \lfloor q/2 \rceil \cdot m'. \tag{1.3}$$

The hint h' is computed via

$$h' = \langle -\mathbf{s}' || \mathbf{e}', \mathbf{e}_u || \mathbf{r} \rangle + y' \tag{1.4}$$

using the reused independent randomness $\mathsf{r}_0 = (\mathbf{r}, \mathbf{e}_u)$, the corresponding private key $(\mathbf{s}', \mathbf{e}')$ and a fresh dependent randomness $\hat{\mathsf{r}}' = y'$. This technique can naturally extend to multiple hints $(h_i)_{i \in [N]}$ given multiple $(\mathbf{b}_i, \mathbf{s}_i)$ and y_i. As a result, we obtain the *reproduction* algorithm and *hint generation* algorithm.

Since the hints $(h_i)_{i \in [N]}$ reveal partial information about the randomness $(\mathbf{r}, \mathbf{e}_u)$, establishing semantic security of the ciphertext is non-trivial. To address this challenge, we rely on the Matrix Hint-MLWE assumption [29] as an intermediary, rather than reducing directly to the standard MLWE assumption. More precisely, the intermediary allows us to precisely measure how much information on the randomness (i.e., the MLWE secret) is leaked from the hints and to make that impact on the hardness of MLWE ciphertext negligible under suitable parameter setting.

To adapt this approach to the lattice-based XR-PKE, we first need to instantiate the Matrix Hint-MLWE assumption. Informally, the Matrix Hint-MLWE assumption states that given a hint vector $\mathbf{h} \in \mathcal{R}^\ell$ where $\mathbf{h} := \mathbf{R}\hat{\mathbf{r}} + \mathbf{y}$, the MLWE instance $[\mathbf{I}|\mathbf{A}]\hat{\mathbf{r}}$ is still indistinguishable from the uniformly random values if $\hat{\mathbf{r}}$ and $\mathbf{y}$ are sampled from appropriate discrete Gaussian distributions. Here, the hint $\mathbf{h}$ in the Matrix Hint-MLWE assumption is composed of the matrix product of an MLWE secret vector $\hat{\mathbf{r}}$ and a bounded matrix $\mathbf{R}$ picked by the adversary, masked by a fresh vector $\mathbf{y}$.

From our intuition in XR-PKE, the hints are in the form of $h_i := \langle \boldsymbol{\gamma}_i, \hat{\mathbf{r}} \rangle + y_i$ for $i \in [N]$. Here, h_i is composed of the inner product of an MLWE secret vector $\hat{\mathbf{r}} := (y || \mathbf{e}_u || \mathbf{r})$ and a vector $\boldsymbol{\gamma}_i := (0 || -\mathbf{s}_i || \mathbf{e}_i)$, which is bounded by $\|\boldsymbol{\gamma}_i\|_\infty \leqslant \nu$ and chosen by the adversary, and masked by a fresh element y_i. Then, we concatenate the hints $(h_i)_{i \in [N]}$ as a hint vector $\mathbf{h}$ such that $\mathbf{h} := \mathbf{R}\hat{\mathbf{r}} + \mathbf{y}$ where $\mathbf{R} := (\boldsymbol{\gamma}_i^\top)_{i \in [N]}$ and $\mathbf{y} := (y_i)_{i \in [N]}$. Here, we get the instantiation of Matrix Hint-MLWE for XR-PKE.

Then, we refine the reduction of Matrix Hint-MLWE from standard MLWE and derive new parameter conditions, as presented in Theorem 4.3. To satisfy such conditions, we carefully choose the distributions for the randomness $(\mathbf{r}, \mathbf{e}_u) \leftarrow \mathcal{D}_{\sigma_0}^n \times \mathcal{D}_{\sigma_0}^m$ and the randomness $y \leftarrow \mathcal{D}_{\sigma_1}$ where $\mathcal{D}_{\sigma_0}$ and $\mathcal{D}_{\sigma_1}$ denote discrete Gaussian distributions with different widths, as opposed to uniform distribution on intervals as used in Kyber, which seems to preclude an efficient Matrix Hint-MLWE to standard MLWE security reduction. More details are provided in Section 4.1 and Section 4.5.

Finally, we employ the reconciliation mechanism from [57] and the bit-dropping technique as in Kyber [14] to compress the ciphertext, particularly the key-dependent ciphertexts, as much as possible. These optimizations bring the bandwidth cost of our mmPKE construction close to *optimal*. In addition, following

the framework of DLP IBE [25] and leveraging the pre-image sampling algorithm in NTRU lattices [20], we extend our construction to obtain lattice-based mmIBE and mmIB-KEM.

2 Preliminaries

In this section, we provide some of the preliminaries needed for our paper, and refer the reader to Appendix A for more preliminaries.

2.1 Notation

Let $\lambda \in \mathbb{N}$ denote the security parameter. For a positive integer n, we denote the set $\{0, \ldots, n-1\}$ by $[n]$ and the set $\{\ell, \ldots, n-1\}$ by $[\ell : n]$. For a positive integer q, we denote $\mathbb{Z}_q$ as the integers modulo q and $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^d+1)$ as the polynomials modulo q and X^d+1. For positive integer ν, we write $\mathbb{S}_\nu$ to denote the set of polynomials in $\mathcal{R}_q$ with infinity norm bounded by ν. The size of the $\mathbb{S}_\nu$ coefficient support is denoted $\bar{\nu} \leqslant 2\nu + 1$; for example $\nu = 1, \bar{\nu} = 2$ indicates binary polynomials. We denote bold lowercase letters as vectors of polynomial elements, e.g., $\mathbf{u} \in \mathcal{R}_q^m$, bold uppercase letters as matrices of polynomial elements, e.g., $\mathbf{U} \in \mathcal{R}_q^{m \times n}$, lowercase letters with an arrow as vectors of integers or reals, e.g., $\vec{a} \in \mathbb{Z}_q^m$, and uppercase letters as matrices of integers or reals, e.g., $A \in \mathbb{R}_q^{m \times n}$. For a polynomial element, e.g., $a \in \mathcal{R}_q$, we define its negacyclic matrix as $\bar{A} := \mathit{\Gamma}(a) \in \mathbb{Z}_q^{d \times d}$. Similarly, for a polynomial vector and matrix, e.g., $\mathbf{b} \in \mathcal{R}^m$ and $\mathbf{D} \in \mathcal{R}_q^{m \times n}$, we define their negacyclic matrix as $\bar{B} := \mathit{\Gamma}(\mathbf{b}) \in \mathbb{Z}^{md \times d}$ and $\bar{D} := \mathit{\Gamma}(\mathbf{D}) \in \mathbb{Z}_q^{md \times nd}$, respectively, where each polynomial element in the vector and matrix is replaced by its negacyclic matrix. For the vectors over integers and polynomials, we denote their inner product as $\langle \cdot, \cdot \rangle$, e.g., $\langle \vec{a}, \vec{b} \rangle$ and $\langle \mathbf{a}, \mathbf{b} \rangle$. We denote $\mathsf{poly}(\lambda)$ as polynomial functions such that $\mathsf{poly}(\lambda) = \bigcup_{c \in \mathbb{N}} O(\lambda^c)$ and $\mathsf{neg}(\lambda)$ as negligible functions such that $\mathsf{neg}(\lambda) = \bigcap_{c \in \mathbb{N}} o(\lambda^{-c})$.

We denote rounding operation as $\lfloor \cdot \rceil$, e.g., $\lfloor a \rceil$ rounds the result to the nearest integer of a. We denote assignment as $:=$, e.g., $x := y$ assigns the value of y to x. We denote sampling or output as $\leftarrow$, e.g., $x \leftarrow \mathcal{D}$ indicates that x is sampled from the distribution $\mathcal{D}$, and $x \leftarrow \mathsf{A}(y)$ denotes that x is the output of probabilistic polynomial time (PPT) algorithm A given input y. Particularly, we write $x \leftarrow S$ when $x \in S$ is sampled uniformly randomly from the finite set S. We denote the uniform distribution on a set S as $\mathcal{U}(S)$. For a vector $\mathbf{a}$ (or $\vec{a}$), we write $\|\mathbf{a}\|$, $\|\mathbf{a}\|_1$, and $\|\mathbf{a}\|_\infty$ to denote its ℓ_2-norm, ℓ_1-norm and ℓ_∞-norm, respectively. For a matrix $\mathbf{A}$ (or A), we write $\|\mathbf{A}\|$, $\|\mathbf{A}\|_1$ and $\|\mathbf{A}\|_\infty$ to denote its matrix 2-norm (largest singular value), matrix 1-norm (maximum column ℓ_1-norm), and matrix ∞-norm (maximum row ℓ_1-norm), respectively. We write $\sigma_{\min}(\mathbf{A})$ and $\sigma_{\max}(\mathbf{A})$ to denote the smallest and largest singular values of $\mathbf{A}$, respectively. For any two subset X, Y of some additive group, we define $-X = \{-x : x \in X\}$ and $X + Y = \{x + y : x \in X, y \in Y\}$.

2.2 Reconciliation Mechanism

We recall the reconciliation mechanism proposed by Peikert [57]. At a high level, this mechanism shows that if an element $v \in \mathbb{Z}_q$ (or $v \in \mathcal{R}_q$) is uniformly random, then its rounding value $\lfloor v \rceil_2$ is uniformly random even given its cross rounding value $\langle v \rangle_2$. And others can recover $\lfloor v \rceil_2$ by $\langle v \rangle_2$ and another value w close to v. We illustrate the mechanism by the following lemmas from [57].

Lemma 2.1. *Define the modular rounding function $\lfloor \cdot \rceil_p : \mathbb{Z}_q \to \mathbb{Z}_p$ as $\lfloor v \rceil_p := \lfloor \frac{p}{q} \cdot v \rceil$ and similar for $\lfloor \cdot \rceil_p$. Define the cross-rounding function $\langle \cdot \rangle_2 : \mathbb{Z}_q \to \mathbb{Z}_2$ as $\langle v \rangle_2 := \lfloor \frac{4}{q} \cdot v \rfloor \mod 2$. Define the randomized function $\mathsf{dbl}(\cdot) : \mathbb{Z}_q \to \mathbb{Z}_{2q}$ as $\mathsf{dbl}(v) := 2v - \bar{e} \in \mathbb{Z}_{2q}$ where $v \in \mathbb{Z}_q$ is an input and e is a error independently sampled from the distribution of a set $\{0, \pm 1\}$ with probability $1/2$, $1/4$, and $1/4$ respectively. For an (odd) modulus q, if $v \in \mathbb{Z}_q$ is uniformly random and $\bar{v} := \mathsf{dbl}(v) \in \mathbb{Z}_{2q}$, then $\lfloor \bar{v} \rceil_2$ is uniformly random even given $\langle \bar{v} \rangle_2$.*

Lemma 2.2. *Define two disjoint intervals as $I_0 := \{0, 1, ..., \lfloor \frac{p}{4} \rceil - 1\}$, $I_1 := \{-\lfloor \frac{p}{4} \rfloor, ..., -1\} \mod p$. Observe that: (1) these intervals from a partition of all the elements $v \in \mathbb{Z}_p$ such that $\lfloor v \rceil_2 = 0$. Similarly $I_0 + \frac{p}{2}$ and $I_1 + \frac{p}{2}$ partition all the elements $v \in \mathbb{Z}_p$ such that $\lfloor v \rceil_2 = 1$; (2) $b = \langle v \rangle_2$ if and only if $v \in I_b \cup (\frac{p}{2} + I_b)$. Define the reconciliation function $\mathsf{rec}(\cdot, \cdot) : \mathbb{Z}_p \times \mathbb{Z}_2 \to \mathbb{Z}_2$ as*

$$\mathsf{rec}(w, b) := \begin{cases} 0 & \text{if } w \in I_b + E \mod p \\ 1 & \text{otherwise.} \end{cases}$$

where the set $E := [-\frac{p}{8}, \frac{p}{8}) \cap \mathbb{Z}$. For even modulus p, if $w = v + e \mod p$ for some $v \in \mathbb{Z}_p$ and $e \in E$, then $\mathsf{rec}(w, \langle v \rangle_2) = \lfloor v \rceil_2$.

Remark 2.3. We can directly extend Lemma 2.1 and 2.2 to polynomial rings $\mathcal{R}_q$ by applying $\lfloor \cdot \rceil_2$, $\langle \cdot \rangle_2$, $\mathsf{dbl}(\cdot)$, $\mathsf{rec}(\cdot, \cdot)$ in coefficient-wise.

2.3 Lattice Preliminaries

We begin with the definition of the standard lattice-based problem. Additional lattice preliminaries are given in Appendix A.1.

Definition 2.4 (MLWE Problem). Let $m, n > 0$ be positive integers. Let χ be an error distribution over $\mathcal{R}^{m+n}$, $\mathbf{A} \leftarrow \mathcal{U}(\mathcal{R}_q^{m \times n})$. Let $\mathbf{r} \leftarrow \chi$ be a secret vector and $\mathbf{u} \leftarrow \mathcal{U}(\mathcal{R}_q^m)$ be a uniformly random vector. The MLWE problem, denoted by $\mathsf{MLWE}_{\mathcal{R},m,n,q,\chi}$, asks an adversary $\mathcal{A}$ to distinguish between $(\mathbf{A}, [\mathbf{I}_m | \mathbf{A}]\mathbf{r})$ and $(\mathbf{A}, \mathbf{u})$. We say $\mathsf{MLWE}_{\mathcal{R},m,n,q,\chi}$ is hard if for any PPT adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ is negligible in λ, i.e.,

$$\begin{aligned} \mathsf{Adv}^{\mathsf{MLWE}}_{\mathsf{para},\mathcal{A}}(\lambda) := \Bigg| & \Pr\left[b = 1 \,\middle|\, \begin{array}{c} \mathbf{A} \leftarrow \mathcal{U}(\mathcal{R}_q^{m \times n}), \mathbf{r} \leftarrow \chi \\ b \leftarrow \mathcal{A}(\mathbf{A}, [\mathbf{I}_m | \mathbf{A}]\mathbf{r}) \end{array} \right] \\ & - \Pr\left[b = 1 \,\middle|\, \begin{array}{c} \mathbf{A} \leftarrow \mathcal{U}(\mathcal{R}_q^{m \times n}), \mathbf{u} \leftarrow \mathcal{U}(\mathcal{R}_q^m) \\ b \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{u}) \end{array} \right] \Bigg| \end{aligned}$$

where $\mathsf{para} = (\mathcal{R}, m, n, q, \chi)$.

2.4 Multi-Message Multi-Recipient Public Key Encryption

Basically, an mmPKE scheme allows a sender to encrypt a set of messages to a set of public keys. We generalize the definition of decomposable mPKE in [39] to mmPKE as follows.

Like [39], our definition of mmPKE can capture all kinds of mmPKE as well, including the non-decomposable mmPKE and the trivial mmPKE constructed from any standard (single recipient) PKE. In the former case, ct_0 is assigned to the multi-recipient ciphertext and all $\widehat{\mathsf{ct}}_i$ are assigned to $\perp$. Towards the latter case, ct_0 is assigned to $\perp$ and $\widehat{\mathsf{ct}}_i$ is assigned to each individual ciphertext.

Definition 2.5 (Decomposable Multi-Message Multi-Recipient PKE). A decomposable mmPKE scheme with a public-private key pair space $\mathcal{K}$, a message space $\mathcal{M}$, a multi-recipient ciphertext space $\mathcal{C}$, and an individual ciphertext space $\mathcal{C}_{\mathsf{s}}$ consists of the following algorithms:

- $\mathsf{pp} \leftarrow \mathsf{mmSetup}(1^\lambda, N)$: On input a security parameter 1^λ and a number of recipients N, it outputs a public parameter pp.
- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{mmKGen}(\mathsf{pp})$: On input a public parameter pp, it outputs a public-private key pair $(\mathsf{pk}, \mathsf{sk}) \in \mathcal{K}$.
- $\mathbf{ct} := (\mathsf{ct}_0, (\widehat{\mathsf{ct}}_i)_{i\in[N]}) \leftarrow \mathsf{mmEnc}(\mathsf{pp}, (\mathsf{pk}_i)_{i\in[N]}, (\mathsf{m}_i)_{i\in[N]}; \mathsf{r}_0, (\hat{\mathsf{r}}_i)_{i\in[N]})$: On input a public parameter pp, N public keys $(\mathsf{pk}_i)_{i\in[N]}$, N messages $(\mathsf{m}_i)_{i\in[N]}$, $(N+1)$ randomness r_0, $(\hat{\mathsf{r}}_i)_{i\in[N]}$, it can be split into two algorithms:
  - $\mathsf{ct}_0 \leftarrow \mathsf{mmEnc}^{\mathsf{i}}(\mathsf{pp}; \mathsf{r}_0)$: On input a public parameter pp, and a randomness r_0, it outputs a public-key-*independent* ciphertext ct_0.
  - $\widehat{\mathsf{ct}}_i \leftarrow \mathsf{mmEnc}^{\mathsf{d}}(\mathsf{pp}, \mathsf{pk}_i, \mathsf{m}_i; \mathsf{r}_0, \hat{\mathsf{r}}_i)$: On input a public parameter pp, a public key pk_i, a message $\mathsf{m}_i \in \mathcal{M}$, and randomness r_0, $\hat{\mathsf{r}}_i$, it outputs a public-key-*dependent* ciphertext $\widehat{\mathsf{ct}}_i$.
- $\mathsf{ct}_i := (\mathsf{ct}_0, \widehat{\mathsf{ct}}_i)/\perp \leftarrow \mathsf{mmExt}(\mathsf{pp}, i, \mathbf{ct})$: On input a public parameter pp, a multi-recipient ciphertext $\mathbf{ct} \in \mathcal{C}$, and an index $i \in \mathbb{N}$, it deterministically outputs the individual ciphertext $\mathsf{ct}_i \in \mathcal{C}_{\mathsf{s}}$ or a symbol $\perp$ to indicate extraction failure.
- $\mathsf{m}/\perp \leftarrow \mathsf{mmDec}(\mathsf{pp}, \mathsf{sk}, \mathsf{ct})$: On input a public parameter pp, a private key sk, and an individual ciphertext $\mathsf{ct} \in \mathcal{C}_{\mathsf{s}}$, it outputs a message $\mathsf{m} \in \mathcal{M}$ or a symbol $\perp$ to indicate decryption failure.

Correctness. We adopt the correctness definition of mmPKE in [5]. Let $\zeta : \mathbb{N} \to [0, 1]$. We say an mmPKE scheme is ζ-correct, if for all $\lambda, N \in \mathbb{N}$ and $i \in [N]$, message $\mathsf{m}_i \in \mathcal{M}$, the following probability holds,

$$\Pr\left[\begin{array}{c|c} \exists i \in [N] : & \mathsf{pp} \leftarrow \mathsf{mmSetup}(1^\lambda, N); \\ \mathsf{mmDec}(\mathsf{pp}, \mathsf{sk}_i, \mathsf{ct}_i) \neq \mathsf{m}_i & \forall i \in [N] : (\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{mmKGen}(\mathsf{pp}); \\ & \mathbf{ct} \leftarrow \mathsf{mmEnc}(\mathsf{pp}, (\mathsf{pk}_i)_{i\in[N]}, (\mathsf{m}_i)_{i\in[N]}); \\ & \mathsf{ct}_i \leftarrow \mathsf{mmExt}(\mathsf{pp}, i, \mathbf{ct}) \end{array}\right] \leqslant \zeta(\lambda).$$

Security. Following [10], we formalize the security model for mmPKE. In contrast to the model in [5], our definition captures *full CPA (or CCA)* security. Briefly, we do not impose the restriction that the two challenge message vectors must have identical structures.

Let mmPKE be an mmPKE scheme, let N, λ be integers. We define the $\mathsf{mmIND\text{-}CPA}^{\mathsf{KOSK}}$ security game in Figure 1 and defer the remaining security models to Appendix A.3, where we also provide a simple extension of our model to the security model in [60].

We say mmPKE is $\mathsf{mmIND\text{-}CPA}^{\mathsf{KOSK}}$ secure if for all PPT adversary $\mathcal{A}$, the following advantage $\mathsf{Adv}^{\mathsf{mmIND\text{-}CPA}^{\mathsf{KOSK}}}_{\mathsf{mmPKE},N,\mathcal{A}}(\lambda)$ is negligible with λ,

$$\left|\Pr[\mathsf{GAME}^{\mathsf{mmIND\text{-}CPA}^{\mathsf{KOSK}}}_{\mathsf{mmPKE},N,\mathcal{A}}(\lambda) = 1] - \frac{1}{2}\right|.$$

We say $\mathcal{A}$ wins if the game outputs 1.

Game $\mathsf{GAME}^{\mathsf{mmIND\text{-}CPA}^{\mathsf{KOSK}}}_{\mathsf{mmPKE},N,\mathcal{A}}(\lambda)$

$(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2) \leftarrow \mathcal{A}$
$\mathsf{pp} \leftarrow \mathsf{mmSetup}(1^\lambda, N)$
$(\ell, \mathsf{st}) \leftarrow \mathcal{A}_0(\mathsf{pp})$
$\forall i \in [\ell]$, $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{mmKGen}(\mathsf{pp})$
$((\mathsf{m}^0_i, \mathsf{m}^1_i)_{i\in[\ell]}, (\mathsf{m}_i)_{i\in[\ell:N]}, (\mathsf{pk}_i, \mathsf{sk}_i)_{i\in[\ell:N]}, \mathsf{st}) \leftarrow \mathcal{A}_1((\mathsf{pk}_i)_{i\in[\ell]}, \mathsf{st})$
req: $\forall i \in [\ell]$, $|\mathsf{m}^0_i| = |\mathsf{m}^1_i|$
req: $\forall i \in [\ell : N]$, $(\mathsf{pk}_i, \mathsf{sk}_i) \in \mathcal{K}$
$b \leftarrow \{0, 1\}$
$\mathbf{ct} \leftarrow \mathsf{mmEnc}(\mathsf{pp}, (\mathsf{pk}_i)_{i\in[N]}, (\mathsf{m}^b_i)_{i\in[\ell]}, (\mathsf{m}_i)_{i\in[\ell:N]})$
$b' \leftarrow \mathcal{A}_2(\mathbf{ct}, \mathsf{st})$
return $[b = b']$

Fig. 1: The $\mathsf{mmIND\text{-}CPA}^{\mathsf{KOSK}}$ security game for mmPKE.

3 Extended Reproducible Public Key Encryption

In this section, we provide the formal definition of XR-PKE, extending on reproducible PKE in [10], and then show how it can be used to build an mmPKE.

Definition 3.1 (XR-PKE). A (decomposable) XR-PKE with a public-private key space $\mathcal{K}$, a message space $\mathcal{M}$, two randomness distributions $(\mathcal{D}_\mathsf{i}, \mathcal{D}_\mathsf{d})$ for key-independent/key-dependent parts, respectively, and a ciphertext space $\mathcal{C}_\mathsf{s}$ consists of the following algorithms:

- $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, N)$: On input a security parameter 1^λ and a reproducibility count N, it outputs a public parameter pp.
- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\mathsf{pp})$: On input a public parameter pp, it outputs a public-private key pair $(\mathsf{pk}, \mathsf{sk}) \in \mathcal{K}$.
- $\mathsf{ct} := (\mathsf{ct}_0, \widehat{\mathsf{ct}}) \leftarrow \mathsf{Enc}(\mathsf{pp}, \mathsf{pk}, \mathsf{m}; \mathsf{r}_0, \hat{\mathsf{r}})$: On input a public parameter pp, a public key pk, a messages m, two randomnesses $(\mathsf{r}_0, \hat{\mathsf{r}})$, it can be split into two algorithms:

- $\mathsf{ct}_0 \leftarrow \mathsf{Enc}^{\mathsf{i}}(\mathsf{pp}; \mathsf{r}_0)$: On input a public parameter pp, and a randomness r_0 sampled from the distribution $\mathsf{r}_0 \leftarrow \mathcal{D}_{\mathsf{i}}$, it outputs a public-key-*independent* ciphertext ct_0.
- $\widehat{\mathsf{ct}} \leftarrow \mathsf{Enc}^{\mathsf{d}}(\mathsf{pp}, \mathsf{pk}, \mathsf{m}; \mathsf{r}_0, \hat{\mathsf{r}})$: On input a public parameter pp, a public key pk, a message $\mathsf{m} \in \mathcal{M}$, and randomness r_0, $\hat{\mathsf{r}}$ where the latter is sampled from distribution $\hat{\mathsf{r}} \leftarrow \mathcal{D}_{\mathsf{d}}$ independently, it outputs a public-key-*dependent* ciphertext $\widehat{\mathsf{ct}}$.

– $\mathsf{m}/\perp \leftarrow \mathsf{Dec}(\mathsf{pp}, \mathsf{sk}, \mathsf{ct})$: On input a public parameter pp, a private key sk, and a ciphertext $\mathsf{ct} \in \mathcal{C}_{\mathsf{s}}$, it outputs a message $\mathsf{m} \in \mathcal{M}$ or a symbol $\perp$ to indicate decryption failure.

– $(\mathsf{h}_i)_{i\in[N]} \leftarrow \mathsf{HintGen}(\mathsf{r}_0, (\mathsf{pk}_i, \mathsf{sk}_i)_{i\in[N]}, (\hat{\mathsf{r}}_i)_{i\in[N]})$: On input a randomness r_0 sampled from the distribution $\mathsf{r}_0 \leftarrow \mathcal{D}_{\mathsf{i}}$, N public-private key pairs $(\mathsf{pk}_i, \mathsf{sk}_i)_{i\in[N]} \in \mathcal{K}$, and N randomnesses $(\hat{\mathsf{r}}_i)_{i\in[N]}$ where each of them is sampled from the distribution $\hat{\mathsf{r}}_i \leftarrow \mathcal{D}_{\mathsf{d}}$ independently, it outputs N hints $(\mathsf{h}_i)_{i\in[N]}$.

– $\mathsf{ct}'/\perp \leftarrow \mathsf{Rep}(\mathsf{ct}, \mathsf{m}', \mathsf{pk}', \mathsf{sk}', \mathsf{h}')$: On input a ciphertext $\mathsf{ct} \in \mathcal{C}_{\mathsf{s}}$, a message $\mathsf{m}' \in \mathcal{M}$, a public-private key pair $(\mathsf{pk}', \mathsf{sk}') \in \mathcal{K}$, and an associated hint h', it outputs a reproduced ciphertext ct' or a symbol $\perp$ to indicate reproducibility failure.

Correctness. Let $\zeta : \mathbb{N} \to [0, 1]$. We say a XR-PKE scheme is ζ-correct, if for all $\lambda, N \in \mathbb{N}^+$, the following probability at most $\zeta(\lambda)$,

$$\Pr\left[\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}, \mathsf{ct}) \neq \mathsf{m} \;\middle|\; \begin{array}{c} \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, N); \\ (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\mathsf{pp}), \mathsf{m} \leftarrow \mathcal{M}; \\ (\mathsf{r}_0, \hat{\mathsf{r}}) \leftarrow \mathcal{D}_{\mathsf{i}} \times \mathcal{D}_{\mathsf{d}}; \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pp}, \mathsf{pk}, \mathsf{m}; \mathsf{r}_0, \hat{\mathsf{r}}) \end{array}\right].$$

Extended Reproducibility. We first define extended reproducibility game in Figure 2. We say that PKE is *extended reproducible* if for any $\lambda, N \in \mathbb{N}^+$, there exists PPT algorithms $\mathsf{HintGen}$ and Rep, called *hint-generation* algorithm and *reproduction* algorithm, respectively, such that $\mathsf{Game}^{\mathsf{ext\text{-}repr}}_{\mathsf{PKE},\mathsf{Rep},N}(\lambda)$ always outputs 1. More precisely, the probability of $\Pr[\mathsf{Game}^{\mathsf{ext\text{-}repr}}_{\mathsf{PKE},\mathsf{Rep},N}(\lambda) = 1] = 1$ hold.

Security. To fit the property of extended reproducibility, we modify the $\mathsf{IND\text{-}ATK}$ security of regular PKE to $\mathsf{IND\text{-}ATK}^{\mathsf{XR}}$ for $\mathsf{ATK} = \{\mathsf{CPA}, \mathsf{CCA}\}$. Roughly speaking, we say an XR-PKE is secure if the hints generated by $\mathsf{HintGen}$ would not help the adversary to break the security of the challenge ciphertext.

Specifically, let PKE be an XR-PKE and we provide the security game of PKE in Figure 3. With the game $\mathsf{Game}^{\mathsf{IND\text{-}ATK}^{\mathsf{XR}}}_{\mathsf{PKE},N,b,\mathcal{A}}(\lambda)$, we say PKE is $\mathsf{IND\text{-}ATK}^{\mathsf{XR}}$ secure if for all PPT adversary $\mathcal{A}$, the following advantage $\mathsf{Adv}^{\mathsf{IND\text{-}ATK}^{\mathsf{XR}}}_{\mathsf{PKE},N,\mathcal{A}}(\lambda)$ is negligible with λ,

$$\left|\Pr\left[\mathsf{GAME}^{\mathsf{IND\text{-}ATK}^{\mathsf{XR}}}_{\mathsf{PKE},N,0,\mathcal{A}}(\lambda) = 0\right] - \Pr\left[\mathsf{GAME}^{\mathsf{IND\text{-}ATK}^{\mathsf{XR}}}_{\mathsf{PKE},N,1,\mathcal{A}}(\lambda) = 0\right]\right|.$$

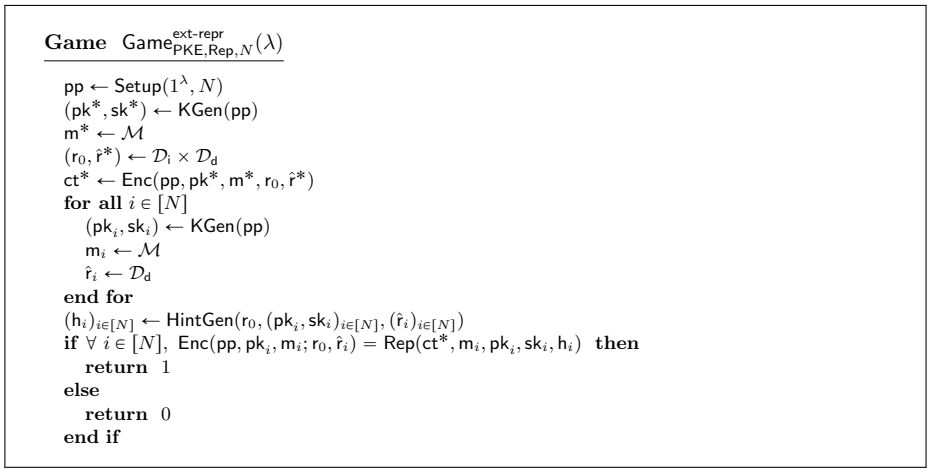

Fig. 2: The extended reproducibility game for PKE.

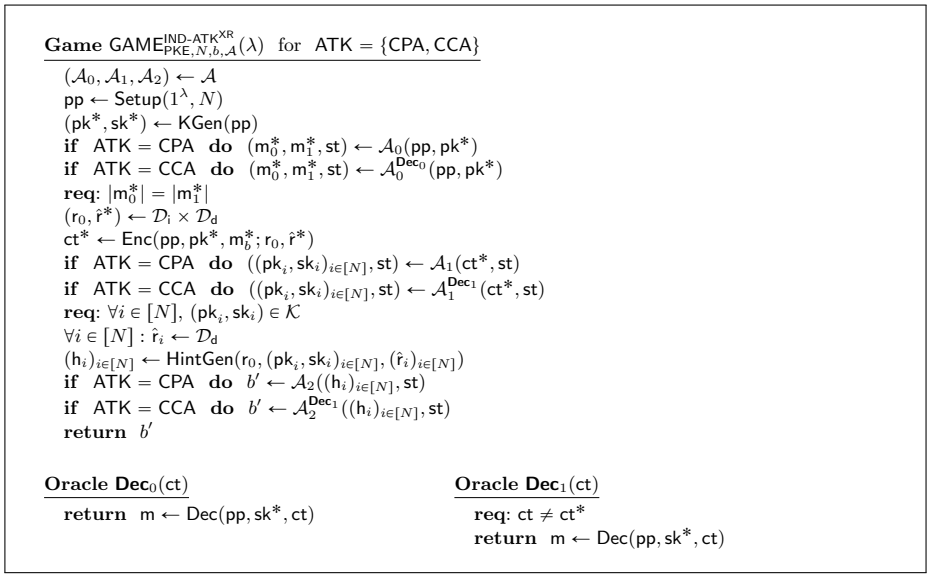

Fig. 3: The $\mathsf{IND\text{-}ATK}^{\mathsf{XR}}$ security game for PKE with $\mathsf{ATK} = \{\mathsf{CPA}, \mathsf{CCA}\}$.

Remark 3.2. Our definition of XR-PKE actually captures the case of original reproducible PKE in [10]. When describing the original reproducible PKE, we can make the hint generation algorithm $\mathsf{HintGen}$ output nothing, i.e., set each of the output hints h_i as an empty symbol $\perp$.

Remark 3.3 (Extension to XR-KEM). We further extend the property of extended reproducibility to the KEM setting, which enables a generic construction

of mmKEM. Due to space constraints, we provide the full definition of XR-KEM, generic construction of mmKEM, along with its security reduction, in Appendix B.

3.1 Generic Construction of mmPKE from XR-PKE

In this subsection, we show the generic construction of mmPKE from XR-PKE.

Construction 3.4 (XR-PKE→mmPKE Compiler). For $\mathsf{ATK} \in \{\mathsf{CPA}, \mathsf{CCA}\}$, let $\mathsf{PKE} = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Enc} = (\mathsf{Enc}^{\mathsf{i}}, \mathsf{Enc}^{\mathsf{d}}), \mathsf{Dec})$ be a (decomposable) $\mathsf{IND\text{-}ATK}^{\mathsf{XR}}$ secure XR-PKE with public-private key space $\mathcal{K}$ and two randomness distributions $(\mathcal{D}_{\mathsf{i}}, \mathcal{D}_{\mathsf{d}})$ for key-independent/key-dependent parts, respectively. Let $\mathsf{Compress}$, $\mathsf{Decompress}$ be the compression and decompression algorithms which can be ignored if there does not exist suitable algorithms. Our compiler $\mathsf{Comp}^{\mathsf{mmPKE}}[\mathsf{PKE}]$ is defined in Figure 4, which outputs an $\mathsf{mmIND\text{-}ATK}^{\mathsf{KOSK}}$ secure mmPKE.

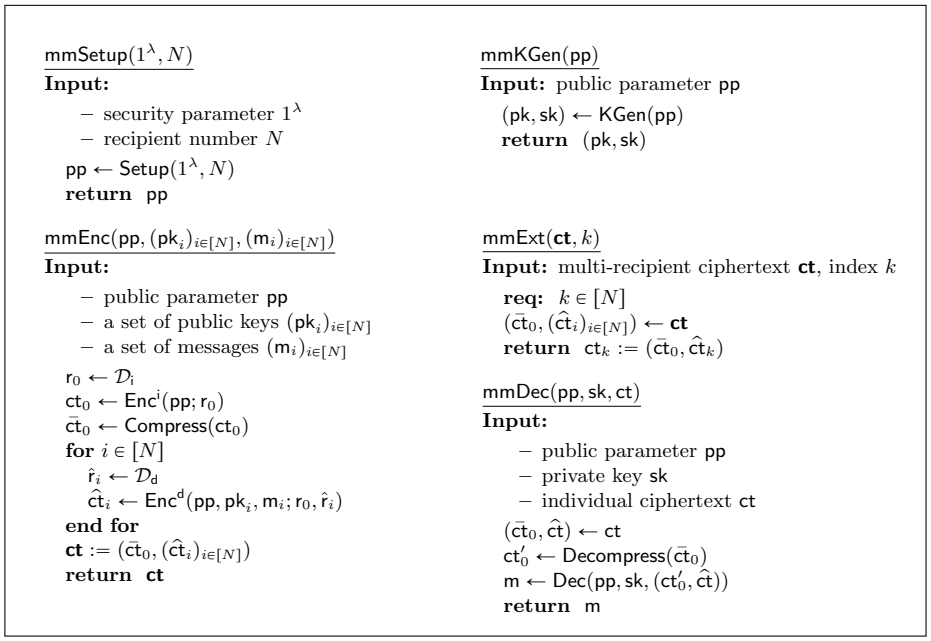

Fig. 4: The $\mathsf{mmIND\text{-}ATK}^{\mathsf{KOSK}}$ mmPKE output by the compiler $\mathsf{Comp}^{\mathsf{mmPKE}}[\mathsf{PKE}]$ for $\mathsf{ATK} \in \{\mathsf{CPA}, \mathsf{CCA}\}$.

Correctness. It is not difficult to see that correctness of our Construction 3.4 follows if the input PKE is correct and the output by decompression algorithm Decompress can still be successfully decrypted with overwhelming probability.

Security. Some intuitive discussion on the security reduction was provided in Technical Overview (Section 1.2). At a high level, since the provided hints do

not help the adversary (or reduction) to break the security of the underlying XR-PKE, we can establish the security of its corresponding mmPKE. Formally, we have the following theorem, the proof of which is given in Appendix G.1.

Theorem 3.5 (Security). *For* $\mathsf{ATK} \in \{\mathsf{CPA}, \mathsf{CCA}\}$*, if* PKE *is* $\mathsf{IND\text{-}ATK}^{\mathsf{XR}}$ *secure and satisfies extended reproducibility, our* $\mathsf{mmPKE} \leftarrow \mathsf{Comp}^{\mathsf{mmPKE}}[\mathsf{PKE}]$ *output by Construction 3.4 is* $\mathsf{mmIND\text{-}ATK}^{\mathsf{KOSK}}$ *secure.*

4 Lattice-Based XR-PKE

In this section, we construct a family of XR-PKEs from lattices which can be used to build efficient mmPKEs via the compiler introduced in the last section.

Our constructions are based on the Matrix Hint-MLWE assumption [29], a variant of the MLWE assumption generalized from the Hint-MLWE assumption [42] and can be reduced from the standard MLWE via appropriate parameters. We first refine this reduction and then give an instantiation for our XR-PKE. Later, we detail the constructions along with their multiple extensions. Finally, we provide the parameter setting and give the theoretical analysis of our mmPKE, compared to the state-of-the-art.

4.1 Refined Matrix Hint-MLWE Assumption

In this subsection, we slightly refine the reduction from standard MLWE to the Matrix Hint-MLWE assumption by introducing a sampleability condition missing in prior works and derive a new parameter setting. We then provide an instantiation of Matrix Hint-MLWE to establish the CPA security of our XR-PKE introduced in the next subsection. We start by recalling the definition of Matrix Hint-MLWE in [29].

Definition 4.1 (Matrix Hint-MLWE, adapted [29]). Let m, n, ℓ be positive integers. Let $\mathcal{S}, \chi_0, \chi_1$ be distributions over $\mathcal{R}^{\ell\times(m+n)}, \mathcal{R}^{m+n}, \mathcal{R}^{\ell}$, respectively. The Matrix Hint-MLWE, denoted by $\mathsf{MatrixHint\text{-}MLWE}^{\ell,\chi_1,\mathcal{S}}_{\mathcal{R},m,n,q,\chi_0}$, asks a PPT adversary $\mathcal{A}$ to distinguish the following two cases:

1. $(\mathbf{A}, [\mathbf{I}_m|\mathbf{A}]\mathbf{r}, \mathbf{R}, \mathbf{h})$ for $\mathbf{A} \leftarrow \mathcal{U}(\mathcal{R}_q^{m\times n})$, $\mathbf{r} \leftarrow \chi_0$, $\mathbf{y} \leftarrow \chi_1$, $\mathbf{R} \leftarrow \mathcal{S}$, and $\mathbf{h} := \mathbf{R}\mathbf{r} + \mathbf{y}$.
2. $(\mathbf{A}, \mathbf{u}, \mathbf{R}, \mathbf{h})$ for $\mathbf{A} \leftarrow \mathcal{U}(\mathcal{R}_q^{m\times n})$, $\mathbf{u} \leftarrow \mathcal{U}(\mathcal{R}_q^{m})$, $\mathbf{r} \leftarrow \chi_0$, $\mathbf{y} \leftarrow \chi_1$, $\mathbf{R} \leftarrow \mathcal{S}$, and $\mathbf{h} := \mathbf{R}\mathbf{r} + \mathbf{y}$.

We say $\mathsf{MatrixHint\text{-}MLWE}^{\ell,\chi_1,\mathcal{S}}_{\mathcal{R},m,n,q,\chi_0}$ is hard if for any PPT adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ is negligible in λ, i.e.,

$$\mathsf{Adv}^{\mathsf{MatrixHint\text{-}MLWE}}_{\mathsf{para},\mathcal{A}}(\lambda) := \left| \Pr\left[b = 1 \;\middle|\; \begin{array}{c} \mathbf{A} \leftarrow \mathcal{U}(\mathcal{R}_q^{m\times n}), \mathbf{r} \leftarrow \chi_0, \mathbf{y} \leftarrow \chi_1, \\ \mathbf{R} \leftarrow \mathcal{S}, \mathbf{h} := \mathbf{R}\mathbf{r} + \mathbf{y}, \\ b \leftarrow \mathcal{A}(\mathbf{A}, [\mathbf{I}_m|\mathbf{A}]\mathbf{r}, \mathbf{R}, \mathbf{h}) \end{array} \right] - \Pr\left[b = 1 \;\middle|\; \begin{array}{c} \mathbf{A} \leftarrow \mathcal{U}(\mathcal{R}_q^{m\times n}), \\ \mathbf{r} \leftarrow \chi_0, \mathbf{y} \leftarrow \chi_1, \mathbf{R} \leftarrow \mathcal{S}, \\ \mathbf{h} := \mathbf{R}\mathbf{r} + \mathbf{y}, \mathbf{u} \leftarrow \mathcal{U}(\mathcal{R}_q^{m}), \\ b \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{u}, \mathbf{R}, \mathbf{h}) \end{array} \right] \right|$$

where $\mathsf{para} = ((\mathcal{R}, m, n, q, \chi_0), (\ell, \chi_1, \mathcal{S}))$.

Here, we slightly adapt the notation towards our needs. Since vector $\mathbf{h}$ contains partial information of the secret vector $\mathbf{r}$, we call $\mathbf{h}$ as hint. Then, we restate the lemma from [29, 42] below which is the stepping-stone to prove the hardness of the Matrix Hint-MLWE assumption. At a high level, the following lemma states that the conditional distribution of $\vec{r}$ given $R\vec{r} + \vec{y}$ turns out to be a non-zero centered skewed Gaussian distribution with a covariance parameter Σ_0 that is dependent on the public matrix R and the covariance parameters of $\vec{r}$ and $\vec{y}$. The restated proof is provided in Appendix G.2.

Lemma 4.2 ([29, 42]). *Let $d, \ell > 0$ be integers. Let Σ_1, $\Sigma_{\mathbf{y}}$ be positive definite symmetric matrices over $\mathbb{R}^{d\times d}$ and $\mathbb{R}^{\ell\times\ell}$, respectively. Let $R \in \mathbb{Z}^{\ell\times d}$ be an integer matrix. Denote $\Sigma_0 := \left(\Sigma_1^{-1} + R^\top \Sigma_{\mathbf{y}}^{-1} R\right)^{-1}$. Then, the following two distributions over $\mathbb{Z}^{d+\ell}$ are statistically identical:*

$$\left\{ \left(\vec{r}, \vec{h}\right) \;\middle|\; \vec{r} \leftarrow \mathcal{D}_{\mathbb{Z}^d, \sqrt{\Sigma_1}},\ \vec{y} \leftarrow \mathcal{D}_{\mathbb{Z}^\ell, \sqrt{\Sigma_{\mathbf{y}}}},\ \vec{h} = R\vec{r} + \vec{y} \right\}$$
$$\approx \left\{ \left(\vec{\hat{r}}, \vec{h}\right) \;\middle|\; \begin{array}{c} \vec{r} \leftarrow \mathcal{D}_{\mathbb{Z}^d, \sqrt{\Sigma_1}},\ \vec{y} \leftarrow \mathcal{D}_{\mathbb{Z}^\ell, \sqrt{\Sigma_{\mathbf{y}}}},\ \vec{h} = R\vec{r} + \vec{y} \\ \vec{c} = \Sigma_0 R^\top {\Sigma_{\mathbf{y}}}^{-1} \vec{h},\ \vec{\hat{r}} \leftarrow \mathcal{D}_{\mathbb{Z}^d, \vec{c}, \sqrt{\Sigma_0}} \end{array} \right\}.$$

Based on the above lemma, we refine the reduction from the standard MLWE to the Matrix Hint-MLWE along with the conditions on the parameters. The proof is provided in Appendix G.2, which presents a refined version of [29].

Theorem 4.3 (Hardness of Matrix Hint-MLWE). *Let m, n, q, ℓ be positive integers. Let $\mathcal{S}$ be a distribution over $\mathcal{R}^{\ell\times(m+n)}$. Let $B > 0$ be a real number such that $\|\bar{R}\|^2 \leqslant B$ where $\bar{R} := \Gamma(\mathbf{R})$ for all possible $\mathbf{R} \leftarrow \mathcal{S}$. Let $\sigma_0, \sigma_1, \sigma, \delta > 0$ be real numbers. Let Σ_1, $\Sigma_{\mathbf{y}}$ be a positive definite symmetric matrices over $\mathbb{R}^{(m+n)d\times(m+n)d}$ and $\mathbb{R}^{\ell d\times \ell d}$, respectively, such that $\|\Sigma_1^{-1}\| \leqslant \frac{1}{\sigma_0^2}$ and $\|\Sigma_{\mathbf{y}}^{-1}\| \leqslant \frac{1}{\sigma_1^2}$. Let $\chi_0 := \mathcal{D}_{\mathbb{Z}^{(m+n)d}, \sqrt{\Sigma_1}}$, $\chi_1 := \mathcal{D}_{\mathbb{Z}^{\ell d}, \sqrt{\Sigma_{\mathbf{y}}}}$, $\chi := \mathcal{D}_{\mathbb{Z}^{(m+n)d}, \sigma}$ be distributions over $\mathcal{R}^{m+n}, \mathcal{R}^\ell, \mathcal{R}^{m+n}$, respectively. There exists an efficient reduction from $\mathsf{MLWE}_{\mathcal{R},m,n,q,\chi}$ to $\mathsf{MatrixHint\text{-}MLWE}^{\ell,\chi_1,\mathcal{S}}_{\mathcal{R},m,n,q,\chi_0}$ that reduces the advantage by at most 2ϵ, if the sampleability condition*

$$\frac{1}{(1+\delta)\sigma^2 + \delta_0} \geqslant \frac{1}{\sigma_0^2} + \frac{B}{\sigma_1^2} \tag{4.1}$$

where $\delta_0 := \sqrt{\frac{\ln(2(m+n)d)+4}{\pi}}$, and the convolution condition

$$\sigma \geqslant \sqrt{1 + 1/\delta} \cdot \eta_\varepsilon(\mathbb{Z}^{(m+n)d}) \tag{4.2}$$

are satisfied.

Specifically, for any PPT adversary $\mathcal{A}$ against the $\mathsf{MatrixHint\text{-}MLWE}^{\ell,\chi_1,\mathcal{S}}_{\mathcal{R},m,n,q,\chi_0}$ assumption, there exists a PPT adversary $\mathcal{B}$ against the $\mathsf{MLWE}_{\mathcal{R},m,n,q,\chi}$ assumption, such that

$$\mathsf{Adv}^{\mathsf{MatrixHint\text{-}MLWE}}_{\mathsf{para}_0,\mathcal{A}}(\lambda) \leqslant \mathsf{Adv}^{\mathsf{MLWE}}_{\mathsf{para}_1,\mathcal{B}}(\lambda) + 2\epsilon$$

where $\mathsf{para}_0 = ((\mathcal{R}, m, n, q, \chi_0), (\ell, \chi_1, \mathcal{S}))$ *and* $\mathsf{para}_1 = (\mathcal{R}, m, n, q, \chi)$.

The above theorem demonstrates the hardness of *elliptic* Matrix Hint-MLWE, a more general case where the secret vector $\mathbf{r}$ and the masking vector $\mathbf{y}$ are sampled from the elliptic discrete Gaussian distributions.

Matrix Hint-MLWE Instantiation for XR-PKE. We first define the distribution $\mathcal{S}$ such that matrix $\mathbf{R}$ can be sampled as follows,

$$\mathbf{R} := \begin{pmatrix} 0 & -\mathbf{s}_0^\top & \mathbf{e}_0^\top \\ \vdots & \vdots & \vdots \\ 0 & -\mathbf{s}_{\ell-1}^\top & \mathbf{e}_{\ell-1}^\top \end{pmatrix} \in \mathcal{R}^{\ell \times (1+m+n)} \tag{4.3}$$

where $\mathbf{s}_i \leftarrow \mathcal{U}(\mathbb{S}_\nu^n)$, $\mathbf{e}_i \leftarrow \mathcal{U}(\mathbb{S}_\nu^m)$ for each $i \in [\ell]$.

Then, we transfer the polynomial matrix $\mathbf{R}$ to its integer matrix $\bar{R} := \Gamma(\mathbf{R}) \in \mathbb{Z}^{\ell d \times (1+m+n)d}$ by substitute the polynomial elements in each vector $\mathbf{s}_i$, $\mathbf{e}_i$ by its negacyclic matrix $\Gamma(\cdot)$ as follows,

$$\bar{R} := \begin{pmatrix} \mathbf{0} & \Gamma(-\mathbf{s}_0) & \Gamma(\mathbf{e}_0) \\ \vdots & \vdots & \vdots \\ \mathbf{0} & \Gamma(-\mathbf{s}_{\ell-1}) & \Gamma(\mathbf{e}_{\ell-1}) \end{pmatrix}.$$

To bound the norm of the matrix $\bar{R}$, we use the inequality $\|\bar{R}\| \leqslant \sqrt{\|\bar{R}\|_1 \cdot \|\bar{R}\|_\infty}$, where $\|\bar{R}\|_1 \leqslant \nu\ell d$ and $\|\bar{R}\|_\infty \leqslant \nu(m+n)d$. Thus, $\|\bar{R}\|^2 \leqslant B$, where

$$B := \ell(m+n)(d\nu)^2 \tag{4.4}$$

Last, we define the matrix $\Sigma_1 \in \mathbb{R}^{(1+m+n)d \times (1+m+n)d}$ and $\Sigma_{\mathbf{y}} \in \mathbb{R}^{\ell d \times \ell d}$ below,

$$\Sigma_1 := \begin{pmatrix} \sigma_1 I_d & \mathbf{0} \\ \mathbf{0} & \sigma_0 I_{(m+n)d} \end{pmatrix}, \quad \Sigma_{\mathbf{y}} := \sigma_1 I_{\ell d}. \tag{4.5}$$

We set $\sigma_1 \geqslant \sigma_0$ so that we have $\|\Sigma_1^{-1}\| = \max(\frac{1}{\sigma_0^2}, \frac{1}{\sigma_1^2}) \leqslant \frac{1}{\sigma_0^2}$ and $\|\Sigma_{\mathbf{y}}^{-1}\| \leqslant \frac{1}{\sigma_1^2}$.

4.2 Construction of XR-PKE

In this subsection, we present the lattice-based construction of XR-PKE. At a high level, we leverage the decryption error as a hint to enable ciphertext reproducibility. In particular, we sample the ciphertext randomness from carefully chosen Gaussian distributions, allowing us to reduce the security of our XR-PKE scheme to the hardness of the Matrix Hint-MLWE problem.

Construction 4.4 (XR-PKE from Lattices). Let λ be a security parameter, $m = m(\lambda)$, $n = n(\lambda)$, $d = d(\lambda)$, $q = q(\lambda)$, $N = N(\lambda)$, $\nu = \nu(\lambda)$ be positive integers. Let $\sigma_0 = \sigma_0(\lambda)$, $\sigma_1 = \sigma_1(\lambda)$ be Gaussian width parameters. For the message space $\mathcal{M} = \{0,1\}^d$, the detailed construction is shown in Figure 5. We summarize the notations in Table 3.

Table 3: Summary of main notations used in our lattice-based XR-PKE/KEM.

Notation	Description
λ	security parameter
ζ	correctness parameter
N	# of recipients
m, n	# of rows of $\mathbf{A}$, # of columns of $\mathbf{A}$
q	system modulus
d	ring dimension of $\mathcal{R} = \mathbb{Z}[X]/(X^d + 1)$
ℓ	dimension of hint vector $\mathbf{h}$ in Matrix Hint-MLWE
ν	ℓ_∞-norm bound on private key $(\mathbf{s}_i, \mathbf{e}_i)$
$\bar{\nu}$	support size $\bar{\nu} \leqslant 2\nu + 1$ of private key $(\mathbf{s}_i, \mathbf{e}_i)$
$\bar{\chi}$	private key distribution
σ_0	Gaussian width of $(\mathbf{r}, \mathbf{e}_u)$ in the ciphertext
χ_1, σ_1	distribution and Gaussian width of y_i in the ciphertext
χ, σ	distribution and Gaussian width of secret in MLWE (hardness equal to Matrix Hint-MLWE)
χ_0, Σ_1	distribution and covariance matrix of secret in Matrix Hint-MLWE
B	square of matrix 2-norm bound on $\bar{R} := \Gamma(\mathbf{R})$
$\mathcal{S}$	distribution of $\mathbf{R}$
d_u	# of bits of each coefficient in key-independent ciphertext $\mathbf{u}$
d_v	# of bits of each coefficient in key-dependent ciphertext u

Extended Reproducibility. We show the extended reproducibility of our construction as follows. The proof is provided in Appendix G.3.

Theorem 4.5 (Extended Reproducibility). *For any positive integer N, our* PKE *in Construction 4.4 is extended reproducible. More precisely, for the extended reproducible game in Figure 2, the probability of* $\Pr[\mathsf{Game}^{\mathsf{ext\text{-}repr}}_{\mathsf{PKE},\mathsf{Rep},N}(\lambda) = 1] = 1$ *holds.*

Correctness. We set $\mathsf{Compress}(x) = \lfloor x \mod q \rceil_{2^{d_u}}$ and $\mathsf{Decompress}(x) = \lfloor x \mod 2^{d_u} \rceil_q$. Here, we mainly consider the case that the (key-independent) ciphertext is compressed and then decompressed before the decryption, as done in mmPKE compiler of Construction 3.4.

We show the correctness of our construction as follows. We will select parameters in Section 4.5 to make our construction ζ-correct with $\zeta \leqslant 2^{-128}$. The proof is provided in Appendix G.3.

Theorem 4.6 (Correctness). *Let $\mathbf{e}, \mathbf{s}, \mathbf{r}, \mathbf{e}_u, y$ be random variables that have the corresponding distribution as in Construction 4.4. Denote ζ as*

$$\Pr\left[\ \left\|\langle \mathbf{e}, \mathbf{r}\rangle + y - \langle \mathbf{s}, \mathbf{e}_u\rangle - c_v + \langle \mathbf{s}, \mathbf{c}_u\rangle\right\|_\infty \geqslant \lfloor q/4 \rceil\ \right]$$

where $\mathbf{c}_u := \mathbf{c} - \lfloor \lfloor \mathbf{c} \bmod q \rceil_{2^{d_u}} \rceil_q \in \mathcal{R}^m$, and $c_v := c - \lfloor \lfloor c \bmod q \rceil_{2^{d_v}} \rceil_q \in \mathcal{R}$. We say our Construction 4.4 is ζ-correct.

Security. We show that our Construction 4.4 is $\mathsf{IND\text{-}CPA}^{\mathsf{XR}}$ secure if the MLWE assumption and the Matrix Hint-MLWE assumption are hard. The proof is provided in Appendix G.3.

$\underline{\mathsf{Setup}(1^\lambda, N)}$
Input:
- security parameter 1^λ
- recipient number N

$\mathbf{A} \leftarrow \mathcal{U}(\mathcal{R}_q^{m\times n})$
return $\mathsf{pp} := \mathbf{A}$

$\underline{\mathsf{KGen}(\mathsf{pp})}$
Input: $\mathsf{pp} = \mathbf{A}$
$(\mathbf{s}, \mathbf{e}) \leftarrow \mathcal{U}(\mathbb{S}_\nu^m) \times \mathcal{U}(\mathbb{S}_\nu^n)$
$\mathbf{b} := \mathbf{A}^\top \mathbf{s} + \mathbf{e}$
return $(\mathsf{pk} := \mathbf{b}, \mathsf{sk} := \mathbf{s})$

$\underline{\mathsf{Enc}(\mathsf{pp}, \mathsf{pk}, \mathsf{m})}$
Input:
- public parameter $\mathsf{pp} = \mathbf{A}$
- public key $\mathsf{pk} = \mathbf{b}$
- message m

$\mathsf{r}_0 := (\mathbf{r}, \mathbf{e}_u) \leftarrow \mathcal{D}_{\sigma_0}^n \times \mathcal{D}_{\sigma_0}^m$
$\mathsf{ct}_0 \leftarrow \mathsf{Enc}^{\mathsf{i}}(\mathsf{pp}; \mathsf{r}_0)$
$\hat{\mathsf{r}} := y \leftarrow \mathcal{D}_{\sigma_1}$
$\widehat{\mathsf{ct}} \leftarrow \mathsf{Enc}^{\mathsf{d}}(\mathsf{pp}, \mathsf{pk}, \mathsf{m}; \mathsf{r}_0, \hat{\mathsf{r}})$
return $\mathsf{ct} := (\mathsf{ct}_0, \widehat{\mathsf{ct}})$

$\underline{\mathsf{Enc}^{\mathsf{d}}(\mathsf{pp}, \mathsf{pk}, \mathsf{m}; \mathsf{r}_0, \hat{\mathsf{r}})}$
Input:
- public parameter $\mathsf{pp} = \mathbf{A}$
- public key $\mathsf{pk} = \mathbf{b}$
- message $\mathsf{m} = m \in \{0,1\}^d$
- randomness $\mathsf{r}_0 = (\mathbf{r}, \mathbf{e}_u)$
- randomness $\hat{\mathsf{r}} = y$

$c := \langle \mathbf{b}, \mathbf{r} \rangle + y + \lfloor \frac{q}{2} \rceil \cdot m$
$u := \lfloor c \mod q \rceil_{2^{d_v}}$
return $\widehat{\mathsf{ct}} := u$

$\underline{\mathsf{Enc}^{\mathsf{i}}(\mathsf{pp}; \mathsf{r}_0)}$
Input:
- public parameter $\mathsf{pp} = \mathbf{A}$
- randomness $\mathsf{r}_0 = (\mathbf{r}, \mathbf{e}_u)$

$\mathbf{c} := \mathbf{A}\mathbf{r} + \mathbf{e}_u$
return $\mathsf{ct}_0 := \mathbf{c}$

$\underline{\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}, \mathsf{ct})}$
Input:
- public parameter $\mathsf{pp} = \mathbf{A}$
- private key $\mathsf{sk} = \mathbf{s}$
- ciphertext $\mathsf{ct} = (\mathbf{c}, u)$

$u' := \lfloor u \mod 2^{d_v} \rceil_q$
$m := \lfloor u' - \langle \mathbf{c}, \mathbf{s} \rangle \mod 2^{d_u} \rceil_2$
return $\mathsf{m} := m$

$\underline{\mathsf{HintGen}(\mathsf{pp}, \mathsf{r}_0, (\mathsf{pk}_i, \mathsf{sk}_i)_{i\in[N]})}$
Input:
- public parameter $\mathsf{pp} = \mathbf{A}$
- randomness $\mathsf{r}_0 = (\mathbf{r}, \mathbf{e}_u)$
- a set of public-private key pairs $(\mathsf{pk}_i, \mathsf{sk}_i)_{i\in[N]} = (\mathbf{b}_i, \mathbf{s}_i)_{i\in[N]}$

for all $i \in [N]$
$y_i \leftarrow \mathcal{D}_{\sigma_1}$
$\mathbf{e}_i := \mathbf{b}_i - \mathbf{A}^\top \mathbf{s}_i$
$h_i := \langle \mathbf{r}, \mathbf{e}_i \rangle - \langle \mathbf{e}_u, \mathbf{s}_i \rangle + y_i$
end for
return $(h_i)_{i\in[N]}$

$\underline{\mathsf{Rep}(\mathsf{ct}, \mathsf{m}', \mathsf{pk}', \mathsf{sk}', \mathsf{h}')}$
Input:
- ciphertext $\mathsf{ct} = (\mathbf{c}, u)$
- message $\mathsf{m}' = m'$
- public-private key $(\mathsf{pk}', \mathsf{sk}') = (\mathbf{b}', \mathbf{s}')$
- hint $\mathsf{h}' = h'$

$c' := \langle \mathbf{c}, \mathbf{s}' \rangle + h' + \lfloor \frac{q}{2} \rceil \cdot m'$
$u' := \lfloor c' \mod q \rceil_{2^{d_v}}$
return $\mathsf{ct}' := (\mathbf{c}, u')$

Fig. 5: An $\mathsf{IND\text{-}CPA}^{\mathsf{XR}}$ secure lattice-based XR-PKE.

Theorem 4.7 (Security). *Let m, n, d, q, N, ν be positive integers parameters. Let $\sigma, \sigma_0, \sigma_1$ be Gaussian width parameters. Let the positive real matrices Σ_1 and $\Sigma_{\mathbf{y}}$ be as Equation (4.5). Let the distribution $\mathcal{S}$ and the bound B be as Equation (4.3) and (4.4) respectively. Let the distribution $\chi_0 := \mathcal{D}_{\mathbb{Z}^{(m+n+1)d}, \sqrt{\Sigma_1}}$, $\chi_1 := \mathcal{D}_{\mathbb{Z}^{Nd}, \sqrt{\Sigma_{\mathbf{y}}}}$, $\bar{\chi} := \mathcal{U}(\mathbb{S}_\nu)$. Suppose Equation (4.1) and (4.2) hold.*

Our PKE *in Construction 4.4 is* $\mathsf{IND\text{-}CPA}^{\mathsf{XR}}$ *secure under the* $\mathsf{MLWE}_{\mathcal{R},n,m,q,\bar{\chi}}$ *and* $\mathsf{MatrixHint\text{-}MLWE}^{N,\chi_1,\mathcal{S}}_{\mathcal{R},m+1,n,q,\chi_0}$ *assumptions. More precisely, for any PPT adversary $\mathcal{A}$, there exist PPT adversaries $\mathcal{B}_0$, $\mathcal{B}_1$ against MLWE assumption and*

Matrix Hint-MLWE assumption, such that

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA}^{\mathsf{XR}}}_{\mathsf{PKE},N,\mathcal{A}}(\lambda) = \mathsf{Adv}^{\mathsf{MLWE}}_{\mathsf{para}_0,\mathcal{B}_0}(\lambda) + \mathsf{Adv}^{\mathsf{MatrixHint\text{-}MLWE}}_{\mathsf{para}_1,\mathcal{B}_1}(\lambda)$$

where $\mathsf{para}_0 := (\mathcal{R}, n, m, q, \bar{\chi})$ *and* $\mathsf{para}_1 := ((\mathcal{R}, m+1, n, q, \chi_0),\ (N, \chi_1, \mathcal{S}))$.

4.3 Construction of XR-KEM

Employing the reconciliation mechanism, we can further compress the ciphertext size, especially for the key-*dependent* ciphertext, and then obtain a lattice-based XR-KEM, which can be used to build an mmKEM. Due to space limitation, we defer its analysis of extended reproducibility, correctness and security in Appendix G.4.

Construction 4.8 (XR-KEM from Lattices). Let λ be a security parameter, $m = m(\lambda)$, $n = n(\lambda)$, $d = d(\lambda)$, $q = q(\lambda)$, $N = N(\lambda)$, $\nu = \nu(\lambda)$ be positive integers. Let $\sigma_0 = \sigma_0(\lambda)$, $\sigma_1 = \sigma_1(\lambda)$ be Gaussian width parameters. Let $\mathsf{dbl}(\cdot)$, $\mathsf{rec}(\cdot,\cdot)$, $\lfloor\cdot\rceil_2$, and $\langle\cdot\rangle_2$ be the functions as define in Lemma 2.1 and Lemma 2.2 which are extended to $\mathcal{R}_q$ per Remark 2.3. For the encapsulated key space $\mathcal{M} = \{0,1\}^d$, the detailed construction is shown in Figure 6. We summarize the notations in Table 3.

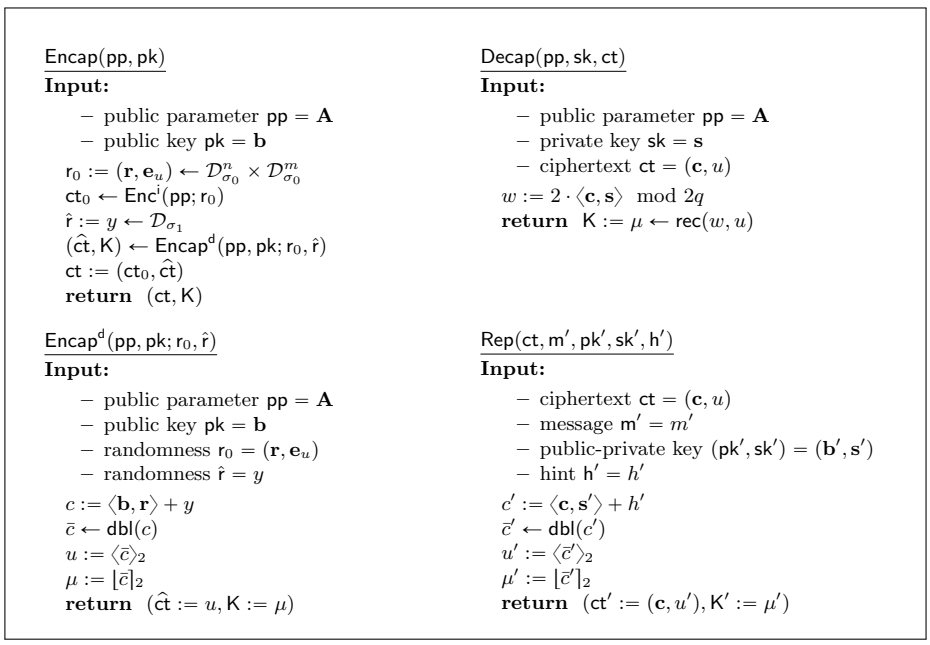

Fig. 6: An $\mathsf{IND\text{-}CPA}^{\mathsf{XR}}$ secure lattice-based XR-KEM where Setup, KGen, $\mathsf{Enc}^{\mathsf{i}}$, and $\mathsf{HintGen}$ are the same as the ones in Construction 4.4.

Remark 4.9 (Extension to Identity-Based Setting). To demonstrate the compatibility of our scheme, we further extend our results to an identity-based setting and obtain the lattice-based mmIBE and mmIB-KEM. At a high level, following the framework of DLP IBE [25], we leverage the pre-image sampling algorithm of NTRU lattices [20] to replace the original setup and key generation algorithms in mmPKE/mmKEM with master key generation and user private key extraction algorithms. Since each user private key is generated by a trusted private key generator (PKG), the KOSK assumption becomes unnecessary in this context. Due to space limitations, we present the formal definition of mmIBE, along with its detailed construction in Appendix F.

4.4 Removing the KOSK Assumption

In this subsection, using a multi-proof extractable NIZK argument system, we present a compiler that can remove the KOSK assumption of the mmPKE with the polynomial-sized number of recipients and provide a detailed analysis of its security. Due to space limitations, we provide an instantiation in Appendix E.1.

Construction 4.10 (KOSK Compiler). For $\mathsf{ATK} \in \{\mathsf{CPA}, \mathsf{CCA}\}$, let mmPKE' be an $\mathsf{mmIND\text{-}ATK}^{\mathsf{KOSK}}$ secure mmPKE with public-private key space $\mathcal{K}$ and randomness distributions $\mathcal{D}_{\mathsf{i}}$, $\mathcal{D}_{\mathsf{d}}$. Let Π be a NIZK argument system. Denote the relation R_Π in Π as

$$R_\Pi := \{(\mathsf{pk}; \mathsf{sk}) \mid (\mathsf{pk}, \mathsf{sk}) \in \mathcal{K}\}$$

We assume the hash value $\mathsf{H}(0) = \mathsf{crs}_\Pi$. The construction of compiler $\mathsf{Comp}^{\mathsf{KOSK}}[\mathsf{mmPKE}', \Pi]$ is defined in Figure 7 which outputs an $\mathsf{mmIND\text{-}ATK}$ secure mmPKE.

The correctness is easy to see. We show how to reduce the security of mmPKE output by $\mathsf{Comp}^{\mathsf{KOSK}}[\mathsf{mmPKE}', \Pi]$ to the security of input mmPKE' and Π. The proof is provided in Appendix G.5.

Theorem 4.11 (Security). *For* $\mathsf{ATK} \in \{\mathsf{CPA}, \mathsf{CCA}\}$*, if* mmPKE' *is* $\mathsf{mmIND\text{-}ATK}^{\mathsf{KOSK}}$ *secure and* Π *is a NIZK argument system satisfies correctness, multi-proof extractability and zero knowledge, our* $\mathsf{mmPKE} \leftarrow \mathsf{Comp}^{\mathsf{KOSK}}[\mathsf{mmPKE}', \Pi]$ *output by Construction 4.10 is* $\mathsf{mmIND\text{-}ATK}$ *secure.*

Remark 4.12 (Recipient Registration and Delegate Verification). In practice, each recipient can be required to "register" to some semi-honest third party, e.g., server in advance. Both proving and the verification for each public key are *one-time* and the latter can be delegated to the server as well. Thus, in this setting, both bandwidth and computation for the encryption do not increase.

4.5 Parameter Setting

In this subsection, we discuss parameter selection for the above constructions. Then, we theoretically demonstrate the performance of the mmPKE/mmKEM built from our constructions, compared to the trivial solution with Kyber.

As discussed before, we need to guarantee that our lattice-based constructions of XR-PKE/KEM satisfy the follow properties:

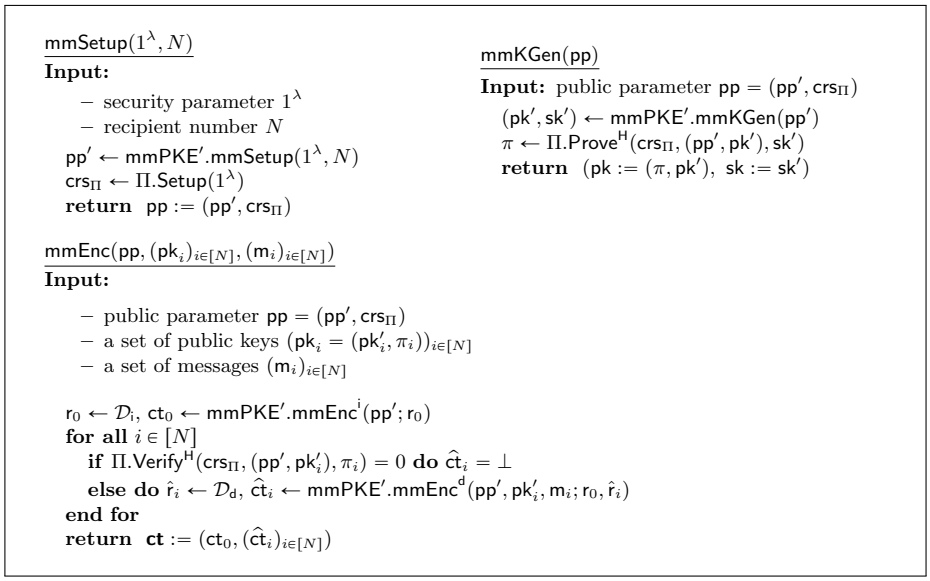

Fig. 7: An mmIND-ATK secure mmPKE output by the compiler $\mathsf{Comp}^{\mathsf{KOSK}}[\mathsf{mmPKE}', \Pi]$ for $\mathsf{ATK} \in \{\mathsf{CPA}, \mathsf{CCA}\}$. mmExt and mmDec are the same as the ones in mmPKE'.

- $\mathsf{MLWE}_{\mathcal{R},n,m,q,\bar{\chi}}$ problem is hard (at 128-bit, 192-bit, and 256-bit security).
- $\mathsf{MatrixHint\text{-}MLWE}^{N,\chi_1,\mathcal{S}}_{\mathcal{R},m+1,n,q,\chi_0}$ problem is hard (at 128-bit, 192-bit, and 256-bit security).
- ζ-correctness holds with $\zeta \leqslant 2^{-128}$.

The parameters of our constructions are summarized in Table 4 and more results are shown in Appendix C.1.

Table 4: Parameter set for our lattice-based constructions of XR-PKE and -KEM, aiming at ζ-correctness with $\zeta \leqslant 2^{-128}$.

N	$\lceil \log q \rceil$	d	m	n	$(\nu, \bar{\nu})$	(d_u, d_v)	(σ_0, σ_1)	pq-sec level
2^{10}	25	256	4	4	$(1, 3)$	$(10, 2)$	$(15.9, 368459)$	128
2^{10}	25	256	7	7	$(1, 2)$	$(11, 2)$	$(15.9, 488797)$	192
2^{10}	25	256	9	9	$(1, 2)$	$(11, 2)$	$(15.9, 554941)$	256

We show how to select the parameters as follows. First, we choose $\nu = 1$, fixing the ℓ_∞-norm of $\mathcal{S}$ and the private key. We choose ternary ($\bar{\nu} = 3$) support $\{0, \pm 1\}$ for $\mathcal{S}$ in the 128-bit parameter set, and binary ($\bar{\nu} = 2$) support $\{0, 1\}$ for 192- and 256-bit parameter sets.

Second, we fix $\delta = 1$ in Theorem 4.3. Then, we need to guarantee that $2\varepsilon \leqslant 2^{-128}$ and the requirements in Equation (4.1) and (4.2) hold. By Lemma A.4, we set $\sigma := \sqrt{2} \cdot \sqrt{\ln(2d(m+n)(1+1/\varepsilon))/\pi}$ so that $\sigma \geqslant \sqrt{2} \cdot \eta_\varepsilon(\mathbb{Z}^{(m+n)d})$

holds. Then, we set $\sigma_0 := 2\sqrt{\sigma^2 + \delta_0/2}$, and $\sigma_1 := 2\sqrt{B}\sqrt{\sigma^2 + \delta_0/2}$ where $\delta_0 := \sqrt{(\ln(2(m+n)d) + 4)/\pi}$ so that $\frac{1}{2\sigma^2+\delta_0} \geqslant \frac{1}{\sigma_0^2} + \frac{B}{\sigma_1^2}$ holds. Here, we set the bound B as in Equation (4.4), i.e., $B := N(m+n)(d\nu)^2$.

Third, we set $n = m$ and $d = 256$. Thus, the encapsulated key space and the short message space $\mathcal{M} = \{0,1\}^{256}$ is the same as the one in Kyber.

Fourth, we pick multiple recipient numbers $N \in \{2^4, 2^7, 2^{10}, 2^{15}\}$ for usability. By Lemma A.3, we can derive the tail bound of the Gaussian distribution to guarantee that the ℓ_∞-norm bound β_{PKE} of the following term in Theorem 4.6 for XR-PKE holds except with negligible probability, i.e., 2^{-128},

$$\beta_{\mathsf{PKE}} := \|\langle \mathbf{e}, \mathbf{r}\rangle + y - \langle \mathbf{s}, \mathbf{e}_u\rangle + \langle \mathbf{s}, \mathbf{c}_u\rangle - c_v\|_\infty < \frac{q}{4}$$

where $(\mathbf{s}, \mathbf{e}) \leftarrow \mathcal{U}(\mathbb{S}_\nu^n) \times \mathcal{U}(\mathbb{S}_\nu^m)$, $(\mathbf{r}, \mathbf{e}_u) \leftarrow \mathcal{D}_{\sigma_0}^n \times \mathcal{D}_{\sigma_0}^m$, $y \leftarrow \mathcal{D}_{\sigma_1}$, $\mathbf{c}_u := \mathbf{c} - \lfloor \lfloor \mathbf{c} \rceil_{2^{d_u}} \rceil_q$, and $c_v := c - \lfloor \lfloor c \rceil_{2^{d_v}} \rceil_q$. Thus, we can bound the ℓ_∞-norms by $\|\mathbf{c}_u\|_\infty \leqslant q/2^{d_u+1}$, and $\|c_v\|_\infty \leqslant q/2^{d_v+1}$, respectively. Similarly, we can derive the tail bound of the ℓ_∞-norm β_{KEM} in Theorem G.8 as well.

Fifth, towards XR-PKE, we fix $d_v = 2$ in advance to compress the size of key-*dependent* ciphertext as much as possible. Note that the sizes of key-*dependent* ciphertext in the constructions of XR-KEM and XR-PKE are both independent with the value of reproducibility count N, i.e., $|\widehat{\mathsf{ct}}| = d/8 = 32$ Bytes and $|\widehat{\mathsf{ct}}| = d \cdot d_v/8 = 64$ Bytes, respectively.

Sixth, we begin by setting the modulus $q \approx 2^{12}$ and $d_u := \lfloor \log q \rfloor$. We compute n, m with $\bar{\chi} := \mathcal{U}(\mathbb{S}_\nu)$ and $\chi := \mathcal{D}_\sigma$ by the LWE estimator [3] to guarantee practical hardness of $\mathsf{MLWE}_{\mathcal{R},n,m,q,\bar{\chi}}$ and $\mathsf{MLWE}_{\mathcal{R},m+N,n,q,\chi}$ at 128-bit, 192-bit, and 256-bit security levels. The latter MLWE assumption stems from $\mathsf{MatrixHint\text{-}MLWE}^{N,\chi_1,\mathcal{S}}_{\mathcal{R},m+N,n,q,\chi_0}$ problem via the reduction in Theorem 4.3. As earlier works [14, 27, 28, 46], we use root Hermite factor (RHF) around 1.0045, 1.0029, 1.0023 to measure the practical hardness of MLWE at 128-bit, 192-bit, and 256-bit secure level, respectively. With the specific n, m, N, q, we compute the ℓ_∞-norm bound β and compare β with $\lceil q/4 \rceil$. We increase the modulus q by factor 2 and repeat computing the parameters until $\beta < \lceil \frac{q}{4} \rceil$.

In the end, after finding the smallest modulus q, we show how to find the smallest d_u in the compression function of mmPKE constructions which can compress the key-independent ciphertext as much as possible. We first change $d_u = 1$ and increase d_u until $\beta < \lceil q/4 \rceil$ holds with overwhelming probability. We provide a script to compute a tight upper bound on ζ as part of our implementation code.

Like the metric in [39], for $\mathsf{CON} \in \{\mathsf{KEM}, \mathsf{PKE}\}$, we use the following formula,

$$k_{\mathsf{com}}^{\mathsf{CON}} := \frac{N \cdot \left|\mathsf{ct}^{\mathsf{Kyber}}\right|}{\left|\mathsf{ct}_0^{\mathsf{CON}}\right| + N \cdot \left|\widehat{\mathsf{ct}}^{\mathsf{CON}}\right|} \xrightarrow{N \to \infty} \frac{\left|\mathsf{ct}^{\mathsf{Kyber}}\right|}{\left|\widehat{\mathsf{ct}}^{\mathsf{CON}}\right|}$$

to measure the *compactness* of our mmPKE/mmKEM compared to the trivial solution via Kyber. One can observe that we achieve significant improvements as $k_{\mathsf{com}}^{\mathsf{KEM}} = 24, 34, 49$ and $k_{\mathsf{com}}^{\mathsf{PKE}} = 12, 17, 24.5$ compared to $\mathsf{Kyber512}$, $\mathsf{Kyber768}$, and $\mathsf{Kyber1024}$ [14], respectively.

5 Implementations and Benchmarks

To evaluate the performance of our constructions, we have implemented the lattice-based mmPKE and mmKEM built from our XR-PKE and XR-KEM, named mmCipher-PKE and mmCipher-KEM, respectively, in portable C [6]. Further details of our implementations and benchmarks are shown in Appendix C.

Benchmarking methodology and baseline. As a baseline comparison, we compare our plain C implementations to the official C reference implementation of (CPA-secure) Kyber[7] with the standard parameter settings of ML-KEM-512, ML-KEM-768, ML-KEM-1024 to achieve 128-bit, 192-bit, 256-bit security, respectively [54, Table 2]. We compare this baseline to our C implementation using the same compiler and target system, an AMD Ryzen 7 4850U Linux laptop running at 3.3 GHz (with overclocking disabled).

To account for memory bandwidth and cache effects, we measure both schemes performing a similar benchmark of encryptions to N recipients, with N growing in powers of two from $N = 1$ up to $N = 1024$. The number of repetitions for each operation is $102400/N$. Average timing is reported.

Encryption/Encapsulation performance. In mmPKE/mmKEM, encryption/encapsulation is the most costly operation as its cost increases with the number of recipients N. Our main contribution is to significantly reduce this cost. We summarize the results on the encryption/encapsulation operation comparing with CPA-secure Kyber (ML-KEM) in Figure 8 and Figure 9.

As predicted by the theoretical analysis in Section 4.5, for $N = 1024$ recipients, among different security levels, mmCipher-KEM and mmCipher-PKE achieve a 23–45× and 12–23× reduction in bandwidth, respectively. In particular, for $N \geqslant 16$ recipients, our constructions already demonstrate a significant improvement (by a factor of over 5). Furthermore, for $N = 1024$ recipients, the bandwidth of our mmCipher-KEM reaches within 4–9% of the plaintext size (*near optimal bandwidth*).

Regarding the computational cost of encapsulation/encryption, for $N = 1024$ recipients, among different security levels, our mmCipher- KEM and mmCipher-PKE offer 3–5× reduction. For $N \geqslant 4$, our constructions are already faster than the baseline. This is because the most expensive operation, i.e., generating the key-independent ciphertext, is amortized across recipients.

Other operations' performance. We list the computational costs of other operations in Table 5. The results show that the key generation and decryption/decapsulation operations are equally fast or faster than those equivalent Kyber operations at the same security level. Note that the input seed (e.g., 32 bytes) for mmSetup() is a public "system parameter" shared by all users, and the operation needs to be re-run only when it changes.

Regarding the bandwidth of key generation, for $N = 1024$ recipients, the public key sizes are 2.4, 4.3, 5.5 KB larger than the one in the baseline, among

[6] Our implementation: https://github.com/ml-kem/mmcipher-artifact
[7] Kyber C reference implementation (`ref`): https://github.com/pq-crystals/kyber

128-, 192-, 256-bit security. However, these additional costs are *one-time* and can be amortized over multiple uses, minimizing their impact on the overall efficiency.

Towards the bandwidth of decryption/decapsulation, for $N = 1024$ recipients, the individual ciphertext in our mmCipher is only 0.5, 1.4, 1.6 KB larger than the one in the baseline, among 128-, 192-, 256-bit security. These additional costs will likely not affect the usability of the scheme in the use cases for which it is best suited.

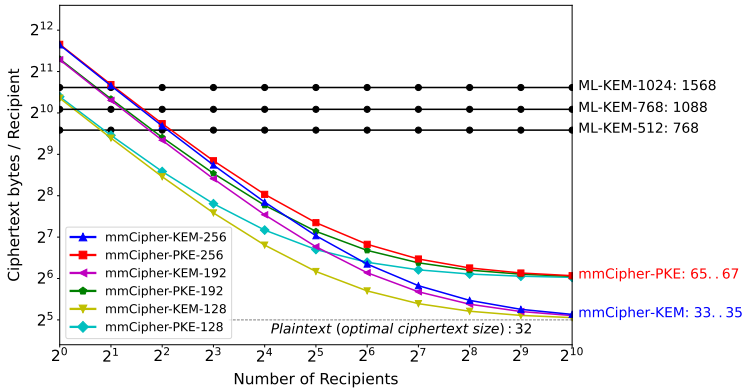

Fig. 8: mmCipher and ML-KEM total ciphertext output in bytes when sending N 256-bit messages (or keys) to N recipients, divided by the number of recipients.

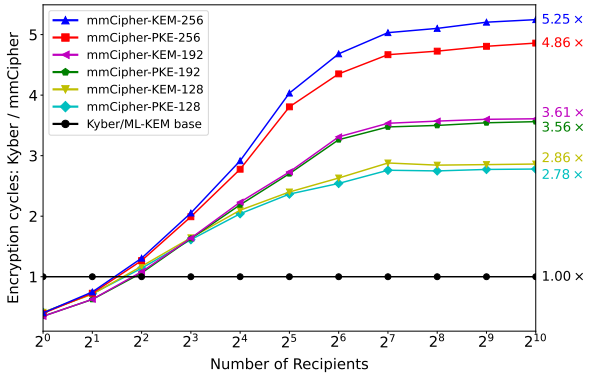

Fig. 9: mmCipher encryption/encapsulation speed when sending N 256-bit messages (or keys) to N recipients, relative to ML-KEM at the same security level.

Table 5: Cycle counts of other operations in mmCipher and ML-KEM (Kyber). Note that K-PKE is an internal CPA subcomponent of ML-KEM.

Operation	*PQ Security*		
	128-bit	**192-bit**	**256-bit**
mmSetup()	188,755	543,640	916,016
mmKGen()	58,815	78,383	106,504
mmDec()	43,511	68,072	85,872
mmDecap()	43,246	67,705	85,323
ML-KEM.KeyGen()	99,145	170,323	262,044
ML-KEM.Decaps()	168,358	259,511	372,644
K-PKE.Decrypt()	40,987	54,547	68,070

References

1. Abou Haidar, C., Passelègue, A., Stehlé, D.: Efficient updatable public-key encryption from lattices. In: Guo, J., Steinfeld, R. (eds.) Advances in Cryptology – ASIACRYPT 2023. pp. 342–373. Springer Nature Singapore, Singapore (2023)
2. Alagic, G., Alagic, G., Apon, D., Cooper, D., Dang, Q., Dang, T., Kelsey, J., Lichtinger, J., Liu, Y.K., Miller, C., et al.: Status report on the third round of the nist post-quantum cryptography standardization process. Tech. rep., National Institute of Standards and Technology (2022)
3. Albrecht, M.R., Player, R., Scott, S.: On the concrete hardness of learning with errors. Journal of Mathematical Cryptology **9**(3), 169–203 (2015)
4. Alwen, J., Hartmann, D., Kiltz, E., Mularczyk, M.: Server-aided continuous group key agreement. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. p. 69–82. CCS '22, Association for Computing Machinery, New York, NY, USA (2022), https://doi.org/10.1145/3548606.3560632
5. Alwen, J., Hartmann, D., Kiltz, E., Mularczyk, M., Schwabe, P.: Post-quantum multi-recipient public key encryption. In: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security. p. 1108–1122. CCS '23, Association for Computing Machinery, New York, NY, USA (2023), https://doi.org/10.1145/3576915.3623185
6. Baek, J., Safavi-Naini, R., Susilo, W.: Efficient multi-receiver identity-based encryption and its application to broadcast encryption. In: Public Key Cryptography-PKC 2005: 8th International Workshop on Theory and Practice in Public Key Cryptography, Les Diablerets, Switzerland, January 23-26, 2005. Proceedings 8. pp. 380–397. Springer (2005)
7. Barbosa, M., Farshim, P.: Randomness reuse: Extensions and improvements. In: Cryptography and Coding: 11th IMA International Conference, Cirencester, UK, December 18-20, 2007. Proceedings 11. pp. 257–276. Springer (2007)
8. Barker, E., Chen, L., Keller, S., Roginsky, A., Vassilev, A., Davis, R.: Recommendation for pair-wise key-establishment schemes using discrete logarithm cryptography. Tech. rep., National Institute of Standards and Technology (2017)
9. Bellare, M., Boldyreva, A., Kurosawa, K., Staddon, J.: Multirecipient encryption schemes: How to save on bandwidth and computation without sacrificing security. IEEE Transactions on Information Theory **53**(11), 3927–3943 (2007)

10. Bellare, M., Boldyreva, A., Staddon, J.: Multi-recipient encryption schemes: Security notions and randomness re-use. In: Desmedt, Y.G. (ed.) Public Key Cryptography — PKC 2003. pp. 85–99. Springer Berlin Heidelberg, Berlin, Heidelberg (2002), https://cseweb.ucsd.edu/~mihir/papers/bbs.pdf
11. Beullens, W., Seiler, G.: Labrador: Compact proofs for r1cs from module-sis. In: Handschuh, H., Lysyanskaya, A. (eds.) Advances in Cryptology – CRYPTO 2023. pp. 518–548. Springer Nature Switzerland, Cham (2023)
12. Boneh, D., Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. SIAM Journal on Computing **36**(5), 1301–1328 (2007)
13. Bootle, J., Lyubashevsky, V., Nguyen, N.K., Sorniotti, A.: A framework for practical anonymous credentials from lattices. In: Handschuh, H., Lysyanskaya, A. (eds.) Advances in Cryptology – CRYPTO 2023. pp. 384–417. Springer Nature Switzerland, Cham (2023)
14. Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Seiler, G., Stehle, D.: Crystals - kyber: A cca-secure module-lattice-based kem. In: 2018 IEEE European Symposium on Security and Privacy (EuroS&P). pp. 353–367. IEEE, London, UK (2018). https://doi.org/10.1109/EuroSP.2018.00032
15. Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors. In: Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing. p. 575–584. STOC '13, Association for Computing Machinery, New York, NY, USA (2013), https://doi.org/10.1145/2488608.2488680
16. Bünz, B., Agrawal, S., Zamani, M., Boneh, D.: Zether: Towards privacy in a smart contract world. In: Bonneau, J., Heninger, N. (eds.) Financial Cryptography and Data Security. pp. 423–443. Springer International Publishing, Cham (2020)
17. Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. In: Advances in Cryptology-EUROCRYPT 2004: International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004. Proceedings 23. pp. 207–222. Springer (2004)
18. Chatterjee, S., Sarkar, P.: Multi-receiver identity-based key encapsulation with shortened ciphertext. In: Barua, R., Lange, T. (eds.) Progress in Cryptology - INDOCRYPT 2006. pp. 394–408. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
19. Cheng, H., Li, X., Qian, H., Yan, D.: Cca secure multi-recipient kem from lpn. In: Naccache, D., Xu, S., Qing, S., Samarati, P., Blanc, G., Lu, R., Zhang, Z., Meddahi, A. (eds.) Information and Communications Security. pp. 513–529. Springer International Publishing, Cham (2018)
20. Chuengsatiansup, C., Prest, T., Stehlé, D., Wallet, A., Xagawa, K.: Modfalcon: Compact signatures based on module-ntru lattices. In: Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. pp. 853–866 (2020)
21. Cramer, R., Shoup, V.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: Krawczyk, H. (ed.) Advances in Cryptology — CRYPTO '98. pp. 13–25. Springer Berlin Heidelberg, Berlin, Heidelberg (1998)
22. Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM Journal on Computing **33**(1), 167–226 (2003)
23. Devoret, M.H., Schoelkopf, R.J.: Superconducting circuits for quantum information: an outlook. Science **339**(6124), 1169–1174 (2013)
24. Diamond, B.E.: Many-out-of-many proofs and applications to anonymous zether. In: 2021 IEEE Symposium on Security and Privacy (SP). pp. 1800–1817. IEEE, San Francisco, CA, USA (2021)

25. Ducas, L., Lyubashevsky, V., Prest, T.: Efficient identity-based encryption over ntru lattices. In: Advances in Cryptology–ASIACRYPT 2014: 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, ROC, December 7-11, 2014, Proceedings, Part II 20. pp. 22–41. Springer (2014)
26. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE transactions on information theory **31**(4), 469–472 (1985)
27. Esgin, M.F., Steinfeld, R., Liu, D., Ruj, S.: Efficient hybrid exact/relaxed lattice proofs and applications to rounding and vrfs. In: Handschuh, H., Lysyanskaya, A. (eds.) Advances in Cryptology – CRYPTO 2023. pp. 484–517. Springer Nature Switzerland, Cham (2023)
28. Esgin, M.F., Steinfeld, R., Zhao, R.K.: MatRiCT+: More efficient post-quantum private blockchain payments. In: 2022 IEEE Symposium on Security and Privacy (SP). pp. 1281–1298. IEEE, USA (2022)
29. Espitau, T., Niot, G., Prest, T.: Flood and submerse: Distributed key generation and robust threshold signature from lattices. In: Reyzin, L., Stebila, D. (eds.) Advances in Cryptology – CRYPTO 2024. pp. 425–458. Springer Nature Switzerland, Cham (2024)
30. Faust, S., Kohlweiss, M., Marson, G.A., Venturi, D.: On the non-malleability of the fiat-shamir transform. In: Progress in Cryptology-INDOCRYPT 2012: 13th International Conference on Cryptology in India, Kolkata, India, December 9-12, 2012. Proceedings 13. pp. 60–79. Springer (2012)
31. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Conference on the theory and application of cryptographic techniques. pp. 186–194. Springer (1986)
32. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In: Wiener, M. (ed.) Advances in Cryptology — CRYPTO' 99. pp. 537–554. Springer Berlin Heidelberg, Berlin, Heidelberg (1999)
33. Gentry, C., Waters, B.: Adaptive security in broadcast encryption systems (with short ciphertexts). In: Joux, A. (ed.) Advances in Cryptology - EUROCRYPT 2009. pp. 171–188. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
34. Guo, Y., Karthikeyan, H., Polychroniadou, A., Huussin, C.: Pride ct: Towards public consensus, private transactions, and forward secrecy in decentralized payments. In: 2024 IEEE Symposium on Security and Privacy (SP). pp. 3904–3922. IEEE (2024)
35. Hashimoto, K., Katsumata, S., Postlethwaite, E., Prest, T., Westerbaan, B.: A concrete treatment of efficient continuous group key agreement via multi-recipient pkes. In: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. p. 1441–1462. CCS '21, Association for Computing Machinery, New York, NY, USA (2021), https://doi.org/10.1145/3460120.3484817
36. Hiwatari, H., Tanaka, K., Asano, T., Sakumoto, K.: Multi-recipient public-key encryption from simulators in security proofs. In: Boyd, C., González Nieto, J. (eds.) Information Security and Privacy. pp. 293–308. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
37. Jain, A., Pandey, O.: Non-malleable zero knowledge: Black-box constructions and definitional relationships. In: Abdalla, M., De Prisco, R. (eds.) Security and Cryptography for Networks. pp. 435–454. Springer International Publishing, Cham (2014)
38. Katsumata, S.: A new simple technique to bootstrap various lattice zero-knowledge proofs to qrom secure nizks. In: Malkin, T., Peikert, C. (eds.) Advances in Cryptology – CRYPTO 2021. pp. 580–610. Springer International Publishing, Cham (2021)

39. Katsumata, S., Kwiatkowski, K., Pintore, F., Prest, T.: Scalable ciphertext compression techniques for post-quantum kems and their applications. In: Moriai, S., Wang, H. (eds.) Advances in Cryptology – ASIACRYPT 2020. pp. 289–320. Springer International Publishing, Cham (2020)
40. Katz, J., Wang, N.: Efficiency improvements for signature schemes with tight security reductions. In: Proceedings of the 10th ACM Conference on Computer and Communications Security. p. 155–164. CCS '03, Association for Computing Machinery, New York, NY, USA (2003), https://doi.org/10.1145/948109.948132
41. Kim, A., Liang, X., Pandey, O.: A new approach to efficient non-malleable zero-knowledge. In: Dodis, Y., Shrimpton, T. (eds.) Advances in Cryptology – CRYPTO 2022. pp. 389–418. Springer Nature Switzerland, Cham (2022)
42. Kim, D., Lee, D., Seo, J., Song, Y.: Toward practical lattice-based proof of knowledge from hint-mlwe. In: Handschuh, H., Lysyanskaya, A. (eds.) Advances in Cryptology – CRYPTO 2023. pp. 549–580. Springer Nature Switzerland, Cham (2023)
43. Kurosawa, K.: Multi-recipient public-key encryption with shortened ciphertext. In: Naccache, D., Paillier, P. (eds.) Public Key Cryptography. pp. 48–63. Springer Berlin Heidelberg, Berlin, Heidelberg (2002)
44. Ledoux, M.: Concentration of measure and isoperimetric inequalities in product spaces. Publications Mathématiques de l'IHÉS **95**(1), 183–206 (2001)
45. Liu, Z., Sotiraki, K., Tromer, E., Wang, Y.: Snake-eye resistance from LWE for oblivious message retrieval and robust encryption. Cryptology ePrint Archive, Paper 2024/510 (2024), https://eprint.iacr.org/2024/510
46. Lyubashevsky, V., Nguyen, N.K., Plançon, M.: Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. In: Dodis, Y., Shrimpton, T. (eds.) Advances in Cryptology – CRYPTO 2022. pp. 71–101. Springer Nature Switzerland, Cham (2022)
47. Lyubashevsky, V., Seiler, G., Steuer, P.: The lazer library: Lattice-based zero knowledge and succinct proofs for quantum-safe privacy. In: Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security. p. 3125–3137. CCS '24, Association for Computing Machinery, New York, NY, USA (2024), https://doi.org/10.1145/3658644.3690330
48. Matsuda, T., Hanaoka, G.: Key encapsulation mechanisms from extractable hash proof systems, revisited. In: Kurosawa, K., Hanaoka, G. (eds.) Public-Key Cryptography – PKC 2013. pp. 332–351. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
49. Micciancio, D., Peikert, C.: Trapdoors for lattices: Simpler, tighter, faster, smaller. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 700–718. Springer (2012)
50. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on gaussian measures. SIAM Journal on Computing **37**(1), 267–302 (2007)
51. Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: Proceedings of the twenty-second annual ACM symposium on Theory of computing. pp. 427–437. Association for Computing Machinery, New York, NY, USA (1990)
52. NIST: SHA-3 standard: Permutation-based hash and extendable-output functions. Federal Information Processing Standards Publication FIPS 202 (August 2015). https://doi.org/10.6028/NIST.FIPS.202
53. NIST: Module-Lattice-Based Digital Signature Standard. Federal Information Processing Standards Publication FIPS 204 (August 2024). https://doi.org/10.6028/NIST.FIPS.204

54. NIST: Module-Lattice-based Key-Encapsulation Mechanism Standard. Federal Information Processing Standards Publication FIPS 203 (August 2024). https://doi.org/10.6028/NIST.FIPS.203
55. Park, J.H., Kim, K.T., Lee, D.H.: Cryptanalysis and improvement of a multi-receiver identity-based key encapsulation at indocrypt 06. In: Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security. p. 373–380. ASIACCS '08, Association for Computing Machinery, New York, NY, USA (2008), https://doi.org/10.1145/1368310.1368366
56. Peikert, C.: An efficient and parallel gaussian sampler for lattices. In: Rabin, T. (ed.) Advances in Cryptology – CRYPTO 2010. pp. 80–97. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
57. Peikert, C.: Lattice cryptography for the internet. In: Mosca, M. (ed.) Post-Quantum Cryptography. pp. 197–219. Springer International Publishing, Cham (2014)
58. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: Proceedings of the fortieth annual ACM symposium on Theory of computing. pp. 187–196 (2008)
59. del Pino, R., Katsumata, S.: A new framework for more efficient round-optimal lattice-based (partially) blind signature via trapdoor sampling. In: Dodis, Y., Shrimpton, T. (eds.) Advances in Cryptology – CRYPTO 2022. pp. 306–336. Springer Nature Switzerland, Cham (2022)
60. Pinto, A., Poettering, B., Schuldt, J.C.: Multi-recipient encryption, revisited. In: Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security. p. 229–238. ASIA CCS '14, Association for Computing Machinery, New York, NY, USA (2014), https://doi.org/10.1145/2590296.2590329
61. Sahai, A.: Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In: 40th annual symposium on foundations of computer science (Cat. No. 99CB37039). pp. 543–553. IEEE, N/A (1999)
62. Smart, N.P.: Efficient key encapsulation to multiple parties. In: Blundo, C., Cimato, S. (eds.) Security in Communication Networks. pp. 208–219. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
63. Steinfeld, R., Ling, S., Pieprzyk, J., Tartary, C., Wang, H.: Ntrucca: How to strengthen ntruencrypt to chosen-ciphertext security in the standard model. In: Public Key Cryptography–PKC 2012: 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, May 21-23, 2012. Proceedings 15. pp. 353–371. Springer (2012)
64. Targhi, E.E., Unruh, D.: Post-quantum security of the fujisaki-okamoto and oaep transforms. In: Hirt, M., Smith, A. (eds.) Theory of Cryptography. pp. 192–216. Springer Berlin Heidelberg, Berlin, Heidelberg (2016)
65. Yang, Z.: On constructing practical multi-recipient key-encapsulation with short ciphertext and public key. Security and Communication Networks **8**(18), 4191–4202 (2015)

Appendix A Additional Preliminaries

A.1 Additional Lattice Preliminaries

Discrete Gaussian Distribution. We first define the n-dimensional spherical Gaussian function $\rho_{\vec{c},\sigma} : \mathbb{R}^n \to (0, 1]$ centered at $\vec{c} \in \mathbb{R}^n$ with a Gaussian width[8]

[8] Note that the Gaussian width σ is related to the standard deviation $\mathfrak{s}$ by $\sigma = \sqrt{2\pi} \cdot \mathfrak{s}$.

$\sigma > 0$ as $\rho_{\vec{c},\sigma}(\vec{x}) := \exp(-\pi \cdot (\vec{x}-\vec{c})^\top(\vec{x}-\vec{c})/\sigma^2)$ for $\vec{x} \in \mathbb{R}^n$. More generally, we define the elliptical Gaussian function $\rho_{\vec{c},\sqrt{\Sigma}} : \mathbb{R}^n \to (0,1]$ centered at $\vec{c} \in \mathbb{R}^n$ with a positive definite symmetric covariance parameter matrix $\Sigma \in \mathbb{R}^{n\times n}$ as $\rho_{\vec{c},\sqrt{\Sigma}}(\vec{x}) := \exp(-\pi \cdot (\vec{x}-\vec{c})^\top \Sigma^{-1}(\vec{x}-\vec{c}))$ for $\vec{x} \in \mathbb{R}^n$. Last, we define the discrete Gaussian distribution $\mathcal{D}_{\Lambda,\Sigma,\vec{c}}$ over an n-dimensional lattice $\Lambda \subseteq \mathbb{R}^n$ centered at $\vec{c}$ with covariance parameter Σ and support Λ as $\mathcal{D}_{\Lambda,\vec{c},\sqrt{\Sigma}} := \frac{\rho_{\vec{c},\sqrt{\Sigma}}(\vec{x})}{\sum_{\vec{y}\in\Lambda}\rho_{\vec{c},\sqrt{\Sigma}}(\vec{y})}$ for $\vec{x} \in \Lambda$. When $\Sigma = \sigma^2\mathbf{I}_n$, i.e., spherical discrete Gaussian distribution, we replace $\sqrt{\Sigma}$ by σ in the subscript and denote it as $\mathcal{D}_{\Lambda,\vec{c},\sigma}$. If $\vec{c} = \vec{0}$, we will omit $\vec{c}$ for simplification.

Lemma A.1 ([15]). *Let $B = (\vec{b}_1, ..., \vec{b}_n)$ be a basis of a full rank n-dimensional lattice Λ, Σ be a positive definite symmetric matrix, $\vec{c} \in \mathbb{R}^n$ be a center, if*

$$\sqrt{\frac{\ln(2n+4)}{\pi}} \cdot \max_i \left\| \Sigma^{-1/2}\vec{b}_i \right\| \leqslant 1$$

holds, there exists a PPT algorithm that can return a sample from $D_{\Lambda,\vec{c},\sqrt{\Sigma}}$.

Lemma A.2 ([42]). *Let Σ_0, Σ_1 be positive definite matrices such that $\Sigma_2^{-1} := \Sigma_0^{-1} + \Sigma_1^{-1}$ satisfies $\sqrt{\Sigma_2} \geqslant \eta_\varepsilon(\mathbb{Z}^n)$ for $0 < \varepsilon < 1/2$. Then for an arbitrary $\vec{c} \in \mathbb{Z}^n$, the distribution*

$$\left\{\vec{x}_0 + \vec{x}_1 | \vec{x}_0 \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sqrt{\Sigma_0}},\ \vec{x}_1 \leftarrow \mathcal{D}_{\mathbb{Z}^n,\vec{c},\sqrt{\Sigma_1}}\right\}$$

is within statistical distance 2ε of $\mathcal{D}_{\mathbb{Z}^n,\vec{c},\sqrt{\Sigma_0+\Sigma_1}}$.

Lemma A.3 ([44]). *For a Gaussian distribution $\mathcal{D}_\sigma$ with Gaussian width $\sigma > 0$, we have $\Pr[|z| \geqslant \tau\cdot\sigma | z \leftarrow \mathcal{D}_\sigma] \leqslant 2\cdot e^{-\pi\tau^2}$. E.g., for $\tau := 5.335$, $2\cdot e^{-\pi\tau^2} \approx 2^{-128}$.*

Smoothing Parameter. As in [50], for an n-dimensional lattice Λ and a positive real $\varepsilon > 0$, the smoothing parameter $\eta_\varepsilon(\Lambda)$ is the smallest s such that $\rho_{1/s}(\Lambda^*\backslash\{\vec{0}\}) \leqslant \varepsilon$ where Λ^* denotes the dual lattice of Λ. As in [56], for a positive definite symmetric matrix Σ, we say $\sqrt{\Sigma} \geqslant \eta_\varepsilon(\Lambda)$ if $\eta_\varepsilon(\sqrt{\Sigma}^{-1}\cdot\Lambda) \leqslant 1$.

Lemma A.4 ([50]). *For any n-dimensional lattice Λ and $\varepsilon > 0$, there exists*

$$\eta_\varepsilon(\Lambda) \leqslant \sqrt{\frac{\ln(2n(1+1/\varepsilon))}{\pi}} \cdot \lambda_n(\Lambda)$$

where $\lambda_n(\Lambda)$ is the smallest real number $r > 0$ such that $\dim(span(\Lambda \cap r\mathcal{B})) = n$ and $\mathcal{B}$ is the n-dimensional unit ball centered at the origin.

Lemma A.5 ([42]). *For a positive definite matrix $\mathbf{\Sigma}$, if $\|\Sigma^{-1}\|_2 \leqslant \eta_\varepsilon(\Lambda)^{-2}$, then $\sqrt{\Sigma} \geqslant \eta_\varepsilon(\Lambda)$.*

A.2 Data Encapsulation Mechanisms

We recall the definition of data encapsulation mechanism (DEM) as follows.

Definition A.6 (Data Encapsulation Mechanisms). A DEM DEM with a key space $\mathsf{DEM}.\mathcal{K} := \{0,1\}^{\mathsf{poly}(\lambda)}$ where λ is the security parameter is defined as follows:

- $\mathsf{c} \leftarrow \mathsf{DEM.Enc}(\mathsf{K}, \mathsf{m})$: Input a key $\mathsf{K} \in \mathsf{DEM}.\mathcal{K}$ and a message $\mathsf{m} \in \{0,1\}^*$, it outputs a ciphertext c.
- $\mathsf{m}/\perp \leftarrow \mathsf{DEM.Dec}(\mathsf{K}, \mathsf{c})$: Input a key $\mathsf{K} \in \mathsf{DEM}.\mathcal{K}$ and a ciphertext c, it outputs a message m or a symbol $\perp$ to indicate decryption failure.

Correctness. A DEM DEM is correct if for all $\mathsf{K} \in \mathsf{DEM}.\mathcal{K}$ and all $\mathsf{m} \in \{0,1\}^*$, we have the following probability holds,

$$\Pr[\mathsf{DEM.Dec}(\mathsf{K}, \mathsf{c}) = \mathsf{m} \mid \mathsf{c} \leftarrow \mathsf{DEM.Enc}(\mathsf{K}, \mathsf{m})] = 1.$$

Security. Let DEM be a DEM scheme, let λ be an integer. We define one-time security of DEM as $\mathsf{OT\text{-}IND\text{-}ATK}$ for $\mathsf{ATK} \in \{\mathsf{CPA}, \mathsf{CCA}\}$ in Figure 10. We say DEM is $\mathsf{OT\text{-}IND\text{-}ATK}$ secure if for all PPT adversary $\mathcal{A}$, the following advantage is negligible with λ,

$$\mathsf{Adv}^{\mathsf{OT\text{-}IND\text{-}ATK}}_{\mathsf{DEM},\mathcal{A}}(\lambda) = \left| \Pr[\mathsf{GAME}^{\mathsf{OT\text{-}IND\text{-}ATK}}_{\mathsf{DEM},\mathcal{A}}(\lambda) = 1] - \frac{1}{2} \right|.$$

We say $\mathcal{A}$ wins if the game outputs 1.

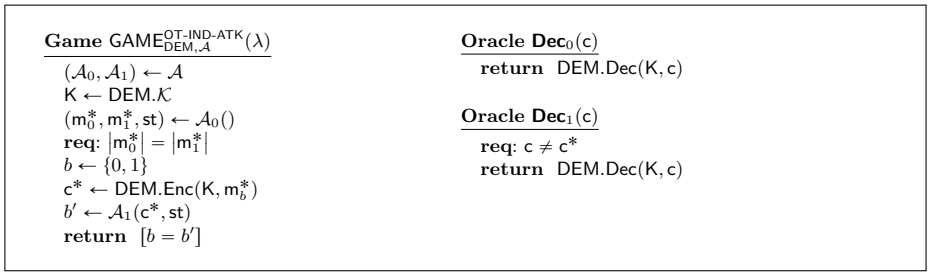

Fig. 10: The $\mathsf{OT\text{-}IND\text{-}ATK}$ security game for DEM. For $\mathsf{ATK} = \mathsf{CCA}$, the adversary $\mathcal{A} := (\mathcal{A}_0, \mathcal{A}_1)$ has the access to the decryption oracles $\mathbf{Dec}_0$, $\mathbf{Dec}_1$ respectively.

A.3 Security Model of Multi-Message Multi-Recipient Public Encryption

Let mmPKE be an mmPKE scheme, let N, λ be integers. Let $\mathsf{ATK} \in \{\mathsf{CPA}, \mathsf{CCA}\}$. We provide multiple security games of mmPKE in Figure 11 to capture different securities of mmPKE.

– $\mathsf{mmIND\text{-}CCA}^{\mathsf{KOSK}}$: We say mmPKE is $\mathsf{mmIND\text{-}CCA}^{\mathsf{KOSK}}$ secure if for all PPT adversary $\mathcal{A}$, the following advantage $\mathsf{Adv}^{\mathsf{mmIND\text{-}CCA}^{\mathsf{KOSK}}}_{\mathsf{mmPKE},N,\mathcal{A}}(\lambda)$ is negligible with λ,

$$\left|\Pr[\mathsf{GAME}^{\mathsf{mmIND\text{-}CCA}^{\mathsf{KOSK}}}_{\mathsf{mmPKE},N,\mathcal{A}}(\lambda)=1]-\frac{1}{2}\right|.$$

We say $\mathcal{A}$ wins if the game outputs 1.

– $\mathsf{mmIND\text{-}ATK}^{\mathsf{Cor}}$: We define $\mathsf{mmIND\text{-}ATK}$ security with adaptive corruption of mmPKE as $\mathsf{mmIND\text{-}ATK}^{\mathsf{Cor}}$. Like [5], we remove the KOSK assumption and give the access of the corruption oracle to the above adversary $\mathcal{A}$. Namely, the adversary $\mathcal{A}$ can adaptively corrupt the recipient by obtaining their private key. To avoid the trivial win, we require that the length of each challenge messages must be the same.

With $\mathsf{GAME}^{\mathsf{IND\text{-}ATK}^{\mathsf{Cor}}}_{\mathsf{mmPKE},N,b,\mathcal{A}}(\lambda)$, we say mmPKE is $\mathsf{mmIND\text{-}ATK}^{\mathsf{Cor}}$ secure if for all PPT adversary $\mathcal{A}$, the following $\mathsf{Adv}^{\mathsf{mmIND\text{-}ATK}^{\mathsf{Cor}}}_{\mathsf{mmPKE},N,\mathcal{A}}(\lambda)$ is negligible with λ,

$$\left|\Pr[\mathsf{GAME}^{\mathsf{mmIND\text{-}ATK}^{\mathsf{Cor}}}_{\mathsf{mmPKE},N,0,\mathcal{A}}(\lambda)=1]-\Pr[\mathsf{GAME}^{\mathsf{mmIND\text{-}ATK}^{\mathsf{Cor}}}_{\mathsf{mmPKE},N,1,\mathcal{A}}(\lambda)=1]\right|.$$

We say $\mathcal{A}$ wins if the game outputs 1.

– $\mathsf{mmIND\text{-}ATK}$: The game of $\mathsf{mmIND\text{-}ATK}$ security for mmPKE is the same as $\mathsf{mmIND\text{-}ATK}^{\mathsf{Cor}}$ except that the adversary cannot obtain the access of corruption oracle. With $\mathsf{GAME}^{\mathsf{IND\text{-}ATK}}_{\mathsf{mmPKE},N,\mathcal{A}}(\lambda)$, we say mmPKE is $\mathsf{mmIND\text{-}ATK}$ secure if for all PPT adversary $\mathcal{A}$, the following advantage $\mathsf{Adv}^{\mathsf{mmIND\text{-}ATK}}_{\mathsf{mmPKE},N,\mathcal{A}}(\lambda)$ is negligible with λ,

$$\left|\Pr[\mathsf{GAME}^{\mathsf{mmIND\text{-}ATK}}_{\mathsf{mmPKE},N,0,\mathcal{A}}(\lambda)=1]-\Pr[\mathsf{GAME}^{\mathsf{mmIND\text{-}ATK}}_{\mathsf{mmPKE},N,1,\mathcal{A}}(\lambda)=1]\right|.$$

We say $\mathcal{A}$ wins if the game outputs 1.

Remark A.7 (Extension to the security model in [60]). The mmPKE security model in [60] is slightly more adaptive than that in [10]. Specifically, in their security model, the adversary $\mathcal{A}$ can see all N public keys of the challenger and adaptively select any subset of them, rather than committing to a fixed number ℓ in advance. Here, we present a simple approach to extending our mmPKE scheme, including [10], to the security model in [60]. This can be achieved by doubling the number of recipients from N to $2N$, under which a reduction from the model in [60] to ours can be established.

Briefly speaking, the reduction first selects $\ell := N$ and obtains N public keys from its challenger, which it then forwards to its adversary. Upon receiving the public keys and the challenge message from the adversary, the reduction identifies the subset of public keys generated by the adversary—say, k of them—and generates the remaining $N-k$ public keys as well as the associated challenge messages. It then sends all $2N$ public keys along with the corresponding challenge messages to its challenger and receives the resulting multi-recipient ciphertext. The reduction extracts and reorders the ciphertext components according to the adversary's specified ordering, and sends the reordered multi-recipient ciphertext

back to the adversary. Finally, the adversary outputs a guess bit, which the reduction forwards to its challenger as its own output.

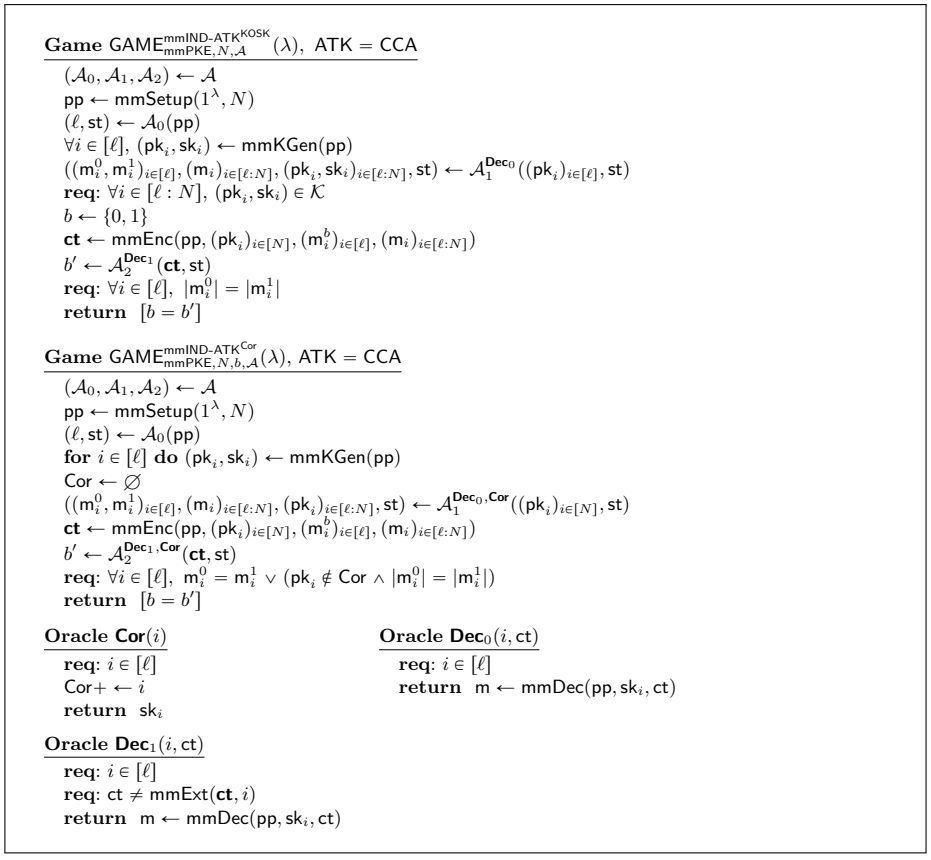

Fig. 11: The $\mathsf{mmIND\text{-}ATK}^{\mathsf{KOSK}}$ and $\mathsf{mmIND\text{-}ATK}^{\mathsf{Cor}}$ security games for mmPKE with $\mathsf{ATK} = \mathsf{CCA}$. For $\mathsf{ATK} = \mathsf{CPA}$, the adversary $\mathcal{A}$ does not have the access of the decryption oracle $\mathbf{Dec}^0$, $\mathbf{Dec}^1$. The mmIND-ATK is the same as $\mathsf{mmIND\text{-}ATK}^{\mathsf{Cor}}$ except that the adversary $\mathcal{A}$ does not have the access of corruption oracle $\mathbf{Cor}$.

A.4 Multi-Key Multi-Recipient Key Encapsulation Mechanism

We generalize the definition of decomposable mmKEM from [60] as follows. Roughly speaking, an mmKEM scheme allows a sender to encapsulate a set of secret (symmetric) keys to a set of recipients. mmKEM can be seen as a special case of mmPKE and it can be transformed into mmPKE for arbitrary length messages with a DEM. We show the transformation at the end of this subsection.

Definition A.8 (Decomposable Multi-Key Multi-Recipient KEM). A decomposable mmKEM scheme over a public-private key pair space $\mathcal{K}$, an encapsulated key space $\mathcal{M}$, a multi-recipient ciphertext space $\mathcal{C}$ and an individual ciphertext space $\mathcal{C}_{\mathsf{s}}$ consists of the following algorithms. The algorithms of $\mathsf{mmSetup}$, mmKGen, mmExt are the same as the ones in Definition 2.5 so we omit them to simplify.

- $(\mathbf{ct} := (\mathsf{ct}_0, (\widehat{\mathsf{ct}}_i)_{i\in[N]}), \mathbf{K} := (\mathsf{K}_i)_{i\in[N]}) \leftarrow \mathsf{mmEncap}(\mathsf{pp}, (\mathsf{pk}_i)_{i\in[N]}; \mathsf{r}_0, (\hat{\mathsf{r}}_i)_{i\in[N]})$: On input a public parameter pp, N public keys $(\mathsf{pk}_i)_{i\in[N]}$, $N+1$ randomness r_0, $(\hat{\mathsf{r}}_i)_{i\in[N]}$, it can be split into two algorithms:
  - $\mathsf{ct}_0 \leftarrow \mathsf{mmEnc}^{\mathsf{i}}(\mathsf{pp}; \mathsf{r}_0)$: On input a public parameter pp, and a randomness r_0, it outputs a public key *independent* ciphertext ct_0.
  - $(\widehat{\mathsf{ct}}_i, \mathsf{K}_i) \leftarrow \mathsf{mmEncap}^{\mathsf{d}}(\mathsf{pp}, \mathsf{pk}_i; \mathsf{r}_0, \hat{\mathsf{r}}_i)$: On input a public parameter pp, a public key pk_i, and randomness r, r_i, it outputs a public key *dependent* ciphertext $\widehat{\mathsf{ct}}_i$ and an encapsulated key $\mathsf{K}_i \in \mathcal{M}$.
- $\mathsf{K}/\perp \leftarrow \mathsf{mmDecap}(\mathsf{pp}, \mathsf{sk}, \mathsf{ct})$: On input a public parameter pp, a private key sk, and an individual ciphertext $\mathsf{ct} \in \mathcal{C}_{\mathsf{s}}$, it outputs the encapsulated key $\mathsf{K} \in \mathcal{M}$ or a symbol $\perp$ to indicate decapsulation failure.

Correctness. Let $\zeta : \mathbb{N} \to [0,1]$. We say an mmKEM scheme is ζ-correct, if for all $\lambda, N \in \mathbb{N}$, the following probability is no more than $\zeta(\lambda)$,

$$\Pr\left[\begin{array}{c} \exists i \in [N] : \\ \mathsf{mmDec}(\mathsf{pp}, \mathsf{sk}_i, \mathsf{ct}_i) \neq \mathsf{K}_i \end{array} \middle| \begin{array}{c} \mathsf{pp} \leftarrow \mathsf{mmSetup}(1^\lambda); \\ \forall i \in [N] : \\ (\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{mmKGen}(\mathsf{pp}); \\ (\mathbf{ct}, (\mathsf{K}_i)_{i\in[N]}) \leftarrow \mathsf{mmEnc}(\mathsf{pp}, (\mathsf{pk}_i)_{i\in[N]}); \\ \mathsf{ct}_i \leftarrow \mathsf{mmExt}(\mathsf{pp}, i, \mathbf{ct}) \end{array} \right].$$

Security. Let mmKEM be an mmKEM scheme, let N, λ be an integer. We provide multiple security games of mmKEM in Figure 12 to capture different securities of mmKEM. Let $\mathsf{ATK} \in \{\mathsf{CPA}, \mathsf{CCA}\}$.

- $\mathsf{mmIND\text{-}ATK}^{\mathsf{KOSK}}$: With $\mathsf{GAME}^{\mathsf{mmIND\text{-}ATK}^{\mathsf{KOSK}}}_{\mathsf{mmKEM},N,\mathcal{A}}(\lambda)$, we say mmKEM is $\mathsf{mmIND\text{-}ATK}^{\mathsf{KOSK}}$ secure if for all PPT adversary $\mathcal{A}$, the following advantage $\mathsf{Adv}^{\mathsf{mmIND\text{-}ATK}^{\mathsf{KOSK}}}_{\mathsf{mmKEM},N,\mathcal{A}}(\lambda)$ is negligible with λ,

$$\left|\Pr[\mathsf{GAME}^{\mathsf{mmIND\text{-}ATK}^{\mathsf{KOSK}}}_{\mathsf{mmKEM},N,\mathcal{A}}(\lambda) = 1] - \frac{1}{2}\right|.$$

  We say $\mathcal{A}$ wins if the game outputs 1.
- $\mathsf{mmIND\text{-}ATK}$: With $\mathsf{GAME}^{\mathsf{IND\text{-}ATK}}_{\mathsf{mmKEM},N,\mathcal{A}}(\lambda)$, we say mmKEM is $\mathsf{mmIND\text{-}ATK}$ secure if for all PPT adversary $\mathcal{A}$, the following advantage $\mathsf{Adv}^{\mathsf{mmIND\text{-}ATK}}_{\mathsf{mmKEM},N,\mathcal{A}}(\lambda)$ is negligible with λ,

$$\left|\Pr[\mathsf{GAME}^{\mathsf{mmIND\text{-}ATK}}_{\mathsf{mmKEM},N,0,\mathcal{A}}(\lambda) = 1] - \Pr[\mathsf{GAME}^{\mathsf{mmIND\text{-}ATK}^{\mathsf{Cor}}}_{\mathsf{mmKEM},N,1,\mathcal{A}}(\lambda) = 1]\right|.$$

  We say $\mathcal{A}$ wins if the game outputs 1.

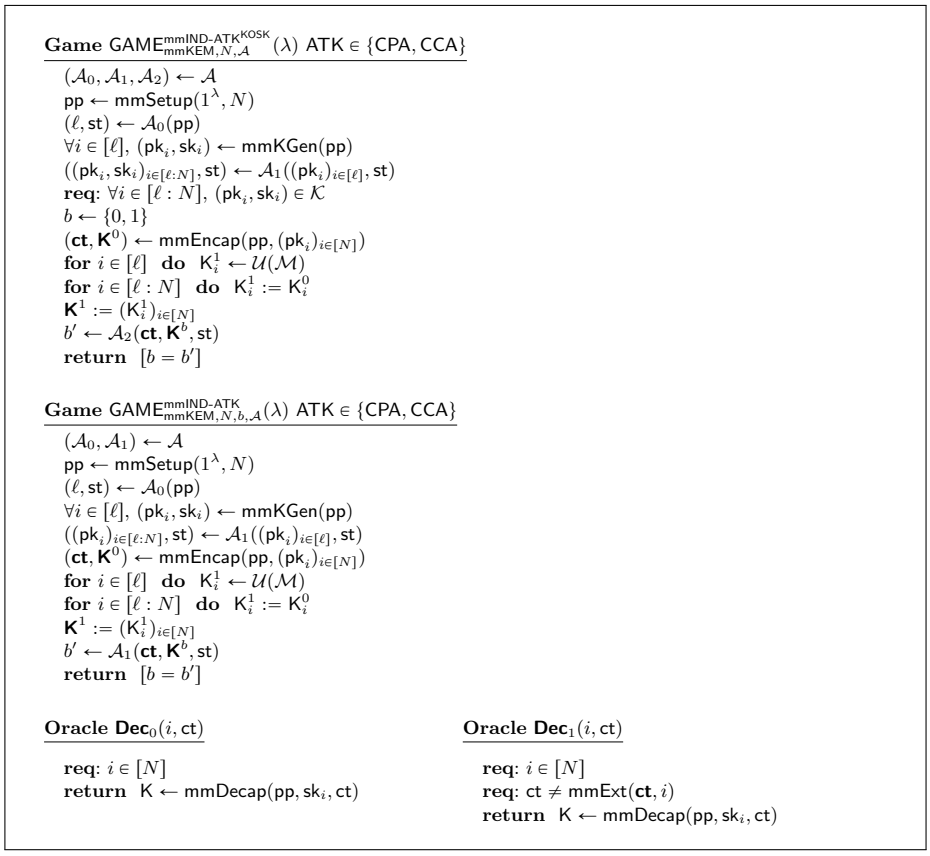

Fig. 12: The $\mathsf{mmIND\text{-}ATK}^{\mathsf{KOSK}}$ and $\mathsf{mmIND\text{-}ATK}$ security games for mmKEM with $\mathsf{ATK} \in \{\mathsf{CPA},\mathsf{CCA}\}$. If $\mathsf{ATK} = \mathsf{CCA}$, the adversary $\mathcal{A}_0$, $\mathcal{A}_1$ can have the access to the decryption oracle $\mathbf{Dec}_0$ and $\mathbf{Dec}_1$ respectively.

Similar with mmPKE, we set the challenge keys as the same for the insider adversary to avoid the trivial win of the adversary $\mathcal{A}$.

Hybrid Encryption. As in [22, 60], the composition of an (mm)KEM and a DEM can yield an (mm)PKE. We recall the detail constructions as follows.

Construction A.9 (Hybrid Encryption Compiler, generalized [60]). For $\mathsf{ATK} \in \{\mathsf{CPA},\mathsf{CCA}\}$, let mmKEM be an $\mathsf{mmIND\text{-}ATK}$ secure (with KOSK assumption) (decomposable) mmKEM, and DEM be an $\mathsf{OT\text{-}IND\text{-}ATK}$ secure DEM such that the (symmetric) key space in mmKEM and DEM are coincide. The construction of compiler $\mathsf{Comp}^{\mathsf{HybE}}[\mathsf{mmKEM},\mathsf{DEM}]$ is defined in Figure 13 which outputs an $\mathsf{mmIND\text{-}ATK}$ secure (with KOSK assumption) mmPKE.

Remark A.10 (Security). We generalize the results in [60] to four security cases, i.e., $\mathsf{mmIND\text{-}ATK}^{\mathsf{KOSK}}$ and $\mathsf{mmIND\text{-}ATK}$ for $\mathsf{ATK} = \{\mathsf{CPA},\mathsf{CCA}\}$, where

they only consider mmIND-CCA case. For the rest cases, the security analysis can be easily derived from the one in [60] so we omit them for simplification.

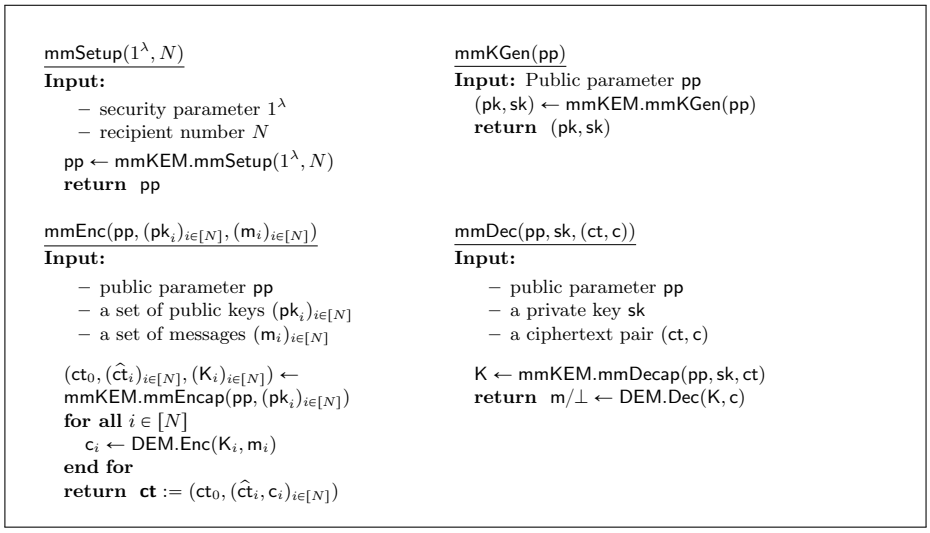

Fig. 13: The mmIND-ATK secure (with KOSK assumption) mmPKE by the compiler $\mathsf{Comp}^{\mathsf{HybE}}[\mathsf{mmKEM}, \mathsf{DEM}]$ for $\mathsf{ATK} \in \{\mathsf{CPA}, \mathsf{CCA}\}$. mmExt with input index i is defined by picking the relevant components $(\mathsf{ct}_i := (\mathsf{ct}_0, \widehat{\mathsf{ct}}_i), \mathsf{c}_i)$ from **ct**.

A.5 Non-Interactive Zero Knowledge Argument System

We recall the definitions of non-interactive zero knowledge (NIZK) argument system in the random oracle from [13, 59] as follows.

Definition A.11 (Non-Interactive Zero Knowledge Argument System). Let R be a polynomial-time verifiable relation of statement-witness (x, w). Denote a language L as a set of statements where there exists a witness w with $(x, w) \in R$. A NIZK protocol Π is defined as follows.

- $\mathsf{crs}_\Pi \leftarrow \Pi.\mathsf{Setup}(1^\lambda)$: On input a security parameter 1^λ, it outputs the common reference string $\mathsf{crs}_\Pi \in \{0,1\}^{\ell(\lambda)}$.
- $\pi/\perp \leftarrow \Pi.\mathsf{Prove}^{\mathsf{H}}(\mathsf{crs}_\Pi, x, w)$: On input the public parameters $\mathsf{crs}_\Pi \in \{0,1\}^\ell$, a statement x and a witness w such that $(x, w) \in R$, it outputs a proof π or an abort symbol $\perp$.
- $0/1 \leftarrow \Pi.\mathsf{Verify}^{\mathsf{H}}(\mathsf{crs}_\Pi, x, \pi)$: On input the public parameters $\mathsf{crs}_\Pi \in \{0,1\}^\ell$, a statement x and a proof π, it output 1 if accepts, otherwise, it outputs 0.

We first define the properties of correctness, zero knowledge, and multi-proof extractability (i.e. straight-line extractability) for NIZK argument system.

Correctness. A NIZK argument system Π is correct if for all $\mathsf{crs}_\Pi \in \{0,1\}^\ell$ and $(x,w) \in R$, the probability that $\Pi.\mathsf{Prove}^{\mathsf{H}}(\mathsf{crs}_\Pi, x, w)$ outputs $\perp$ is $\mathsf{negl}(\lambda)$, and the following probability holds,

$$\Pr\left[\begin{matrix}\pi \leftarrow \Pi.\mathsf{Prove}^{\mathsf{H}}(\mathsf{crs}_\Pi, x, w) : \\ \Pi.\mathsf{Verify}^{\mathsf{H}}(\mathsf{crs}_\Pi, x, \pi) = 1 \big| \pi \neq \perp\end{matrix}\right] = 1 - \mathsf{negl}(\lambda).$$

Zero-Knowledge. A NIZK argument system Π is zero-knowledge if for any PPT adversary $\mathcal{A}$, there exists a simulator $\Pi.\mathsf{Sim} = (\mathsf{Sim}_0, \mathsf{Sim}_1)$ which consists of two PPT algorithms with a shared state such that the following $\mathsf{Adv}^{\mathsf{ZK}}_{\Pi,\mathcal{A}}(\lambda)$ is negligible in λ,

$$\left|\Pr[1 \leftarrow \mathcal{A}^{\mathsf{H},\Pi.\mathsf{Prove}}(\mathsf{crs}_\Pi)] - \Pr[1 \leftarrow \mathcal{A}^{\mathsf{Sim}_0,\mathsf{Sim}_1}(\mathsf{crs}_\Pi)]\right| = \mathsf{negl}(\lambda)$$

where $\Pi.\mathsf{Prove}$ and $\Pi.\mathsf{Sim}$ are prover and simulator oracles which, given (x,w), output $\perp$ if $(x,w) \notin R$ and otherwise return $\Pi.\mathsf{Prove}^{\mathsf{H}}(\mathsf{crs}_\Pi, x, w)$ and $\mathsf{Sim}_1(\mathsf{crs}_\Pi, x)$ respectively. The probability is also taken over the randomness of generating the common reference string $\mathsf{crs}_\Pi \leftarrow \mathsf{Setup}(1^\lambda)$.

Multi-Proof Extractability. A NIZK argument system Π has multi-proof extractability if the following hold:

- *CRS Simulatability:* For any PPT adversary $\mathcal{A}$, we have the following $\mathsf{Adv}^{\mathsf{crs}}_{\Pi,\mathcal{A}}(\lambda)$ is negligible in λ,

$$\begin{aligned}&|\Pr[\mathsf{crs}_\Pi \leftarrow \Pi.\mathsf{Setup}(1^\lambda) : 1 \leftarrow \mathcal{A}^{\mathsf{H}}(\mathsf{crs}_\Pi)] - \\ &\Pr[(\widetilde{\mathsf{crs}}_\Pi, \tau) \leftarrow \mathsf{Sim}_{\mathsf{crs}}(1^\lambda) : 1 \leftarrow \mathcal{A}^{\mathsf{H}}(\widetilde{\mathsf{crs}}_\Pi)]| = \mathsf{negl}(\lambda)\end{aligned}$$

- *Straight-Line Extractability:* There exist constants e_1, e_2, c such that for any $Q_{\mathsf{H}}, Q_s \in \mathsf{poly}(\lambda)$ and any PPT adversary $\mathcal{A}$ that makes at most Q_{H} random oracle queries with

$$\Pr\left[\begin{matrix}(\widetilde{\mathsf{crs}}_\Pi, \tau) \leftarrow \mathsf{Sim}_{\mathsf{crs}}(1^\lambda), \\ \{(x_i, \pi_i)\}_{i\in[Q_s]} \leftarrow \mathcal{A}^{\mathsf{H}}(\widetilde{\mathsf{crs}})\end{matrix} : \begin{matrix}\forall i \in [Q_s], \\ \Pi.\mathsf{Verify}^{\mathsf{H}}(\widetilde{\mathsf{crs}}_\Pi, x_i, \pi_i) = 1\end{matrix}\right] \geqslant \varepsilon(\lambda)$$

  where $\varepsilon(\lambda)$ is non-negligible, we have

$$\Pr\left[\begin{matrix}(\widetilde{\mathsf{crs}}_\Pi, \tau) \leftarrow \mathsf{Sim}_{\mathsf{crs}}(1^\lambda), \{(x_i,\pi_i)\}_{i\in[Q_s]} \leftarrow \mathcal{A}^{\mathsf{H}}(\widetilde{\mathsf{crs}}_\Pi), \\ \{w_i \leftarrow \mathsf{Multi\text{-}Extract}(Q_{\mathsf{H}}, Q_s, 1/\varepsilon, \widetilde{\mathsf{crs}}_\Pi, \tau, x_i, \pi_i)\}_{i\in[Q_s]} : \\ \forall i \in [Q_s], (x_i, w_i) \in R \wedge \mathsf{Verify}^{\mathsf{H}}(\widetilde{\mathsf{crs}}_\Pi, x_i, \pi_i) = 1\end{matrix}\right] \geqslant \frac{1}{2}\cdot\varepsilon(\lambda) - \mathsf{negl}(\lambda)$$

  where the runtime of the extractor is upper-bound by $Q_{\mathsf{H}}^{e_1} \cdot Q_s^{e_2} \cdot \frac{1}{\varepsilon(\lambda)^c} \cdot \mathsf{poly}(\lambda)$.

There are some other scenarios that requires NIZK argument system satisfying other properties. Following [61], we define simulation soundness as follows. Remark that the notion of simulation soundness is a form of non-malleability of NIZK, as noted in [37, 41, 61]. For simplification, here we do not involve the random oracle model.

Simulation Soundness. A NIZK protocol Π is simulation sound if for any PPT adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$, any PPT relations R along with its language L, the following $\mathsf{Adv}^{\mathsf{SS}}_{\Pi,\mathcal{A}}(\lambda)$ is negligible in λ,

$$\Pr\left[\begin{array}{c} \pi \neq \pi'; \\ x' \notin L; \\ \Pi.\mathsf{Verify}(\mathsf{crs}_\Pi, x', \pi') = 1 \end{array} \middle| \begin{array}{c} (\mathsf{crs}_\Pi, \tau) \leftarrow \mathsf{Sim}_0(1^\lambda); \\ (x, \mathsf{st}) \leftarrow \mathcal{A}_0(\mathsf{crs}_\Pi); \\ \pi \leftarrow \mathsf{Sim}_1(\mathsf{crs}_\Pi, x, \tau); \\ (x', \pi') \leftarrow \mathcal{A}_1(\mathsf{crs}_\Pi, x, \pi, \mathsf{st}) \end{array}\right].$$

Appendix B Extended Reproducible Key Encapsulation Mechanism

In this section, we slightly adapt the definition of XR-PKE to obtain the definition of XR-KEM. Then, we show the generic construction of mmKEM from XR-KEM.

Definition B.1 (Extended Reproducible KEM). A (decomposable) extended reproducible KEM with a public-private key space $\mathcal{K}$, an encapsulated key space $\mathcal{M}$, two randomness distributions $(\mathcal{D}_\mathsf{i}, \mathcal{D}_\mathsf{d})$ for key-independent/key-dependent parts, respectively, and a ciphertext space $\mathcal{C}_\mathsf{s}$ consists of the following algorithms. Setup, KGen, $\mathsf{HintGen}$ algorithms are the same as the ones in the definition of XR-PKE and we introduce the rest as follows.

- $(\mathsf{ct}, \mathsf{K}) := (\mathsf{ct}_0, \widehat{\mathsf{ct}}) \leftarrow \mathsf{Encap}(\mathsf{pp}, \mathsf{pk}; \mathsf{r}_0, \hat{\mathsf{r}})$: On input a public parameter pp, a public keys pk, two randomness $(\mathsf{r}_0, \hat{\mathsf{r}})$, it can be split into two algorithms:
  - $\mathsf{ct}_0 \leftarrow \mathsf{Enc}^\mathsf{i}(\mathsf{pp}; \mathsf{r}_0)$: On input a public parameter pp, and a randomness r_0 sampled from the distribution $\mathsf{r}_0 \leftarrow \mathcal{D}_\mathsf{i}$, it outputs a public key *independent* ciphertext ct_0.
  - $(\widehat{\mathsf{ct}}, \mathsf{K}) \leftarrow \mathsf{Encap}^\mathsf{d}(\mathsf{pp}, \mathsf{pk}; \mathsf{r}_0, \hat{\mathsf{r}})$: On input a public parameter pp, a public key pk, and randomness r_0, $\hat{\mathsf{r}}$ where the later is sampled from distribution $\hat{\mathsf{r}} \leftarrow \mathcal{D}_\mathsf{d}$ independently, it outputs a public key *dependent* ciphertext $\widehat{\mathsf{ct}}$ and an encapsulated key $\mathsf{K} \in \mathcal{M}$.
- $\mathsf{K}/\perp \leftarrow \mathsf{Decap}(\mathsf{pp}, \mathsf{sk}, \mathsf{ct})$: On input a public parameter pp, a private key sk, and a ciphertext $\mathsf{ct} \in \mathcal{C}_\mathsf{s}$, it outputs an encapsulated key $\mathsf{K} \in \mathcal{M}$ or a symbol $\perp$ to indicate decapsulation failure.
- $(\mathsf{ct}', \mathsf{K}')/\perp \leftarrow \mathsf{Rep}(\mathsf{ct}, \mathsf{pk}', \mathsf{sk}', \mathsf{h}')$: On input a ciphertext $\mathsf{ct} \in \mathcal{C}_\mathsf{s}$, a public-private key pair $(\mathsf{pk}', \mathsf{sk}') \in \mathcal{K}$, and an associated hint h', it outputs a reproduced ciphertext ct' along with an encapsulated key $\mathsf{K}' \in \mathcal{M}$ or a symbol $\perp$ to indicate reproducibility failure.

Correctness. Let $\zeta : \mathbb{N} \to [0, 1]$. We say a KEM scheme is ζ-correct, if for all $\lambda, N \in \mathbb{N}^+$, the following probability at most $\zeta(\lambda)$,

$$\Pr\left[\mathsf{Decap}(\mathsf{pp}, \mathsf{sk}, \mathsf{ct}) \neq \mathsf{K} \middle| \begin{array}{c} \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, N); \\ (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\mathsf{pp}); \\ (\mathsf{r}_0, \hat{\mathsf{r}}) \leftarrow \mathcal{D}_\mathsf{i} \times \mathcal{D}_\mathsf{d}; \\ (\mathsf{ct}, \mathsf{K}) \leftarrow \mathsf{Encap}(\mathsf{pp}, \mathsf{pk}; \mathsf{r}_0, \hat{\mathsf{r}}) \end{array}\right].$$

Extended Reproducibility. We first define extended reproducibility game for KEM in Figure 14. We say that KEM is *extended reproducible* if for any $\lambda, N \in \mathbb{N}^+$, there exists PPT algorithms $\mathsf{HintGen}$ and Rep, called *hint-generation* algorithm and *reproduction* algorithm, respectively, such that $\mathsf{Game}^{\mathsf{ext\text{-}repr}}_{\mathsf{KEM},\mathsf{Rep},N}(\lambda)$ always outputs 1. More precisely, the following probabilities hold,

$$\Pr\left[\mathsf{Game}^{\mathsf{ext\text{-}repr}}_{\mathsf{KEM},\mathsf{Rep},N}(\lambda) = 1\right] = 1.$$

Game $\mathsf{Game}^{\mathsf{ext\text{-}repr}}_{\mathsf{KEM},\mathsf{Rep},N}(\lambda)$

$\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, N)$
$(\mathsf{pk}^*, \mathsf{sk}^*) \leftarrow \mathsf{KGen}(\mathsf{pp})$
$(\mathsf{r}_0, \hat{\mathsf{r}}^*) \leftarrow \mathcal{D}_\mathsf{i} \times \mathcal{D}_\mathsf{d}$
$(\mathsf{ct}^*, \mathsf{K}^*) \leftarrow \mathsf{Encap}(\mathsf{pp}, \mathsf{pk}^*, \mathsf{r}_0, \hat{\mathsf{r}}^*)$
for all $i \in [N]$
  $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KGen}(\mathsf{pp})$
  $\hat{\mathsf{r}}_i \leftarrow \mathcal{D}_\mathsf{d}$
end for
$(\mathsf{h}_i)_{i\in[N]} \leftarrow \mathsf{HintGen}(\mathsf{r}_0, (\mathsf{pk}_i, \mathsf{sk}_i)_{i\in[N]}, (\hat{\mathsf{r}}_i)_{i\in[N]})$
if $\forall\ i \in [N],\ \mathsf{Encap}(\mathsf{pp}, \mathsf{pk}_i, \mathsf{r}_0, \hat{\mathsf{r}}_i) = \mathsf{Rep}(\mathsf{ct}^*, \mathsf{pk}_i, \mathsf{sk}_i, \mathsf{h}_i)$ **then**
  return 1
else
  return 0
end if

Fig. 14: The extended reproducibility game for KEM.

Security. To fit the property of extended reproducibility, we modify the $\mathsf{IND\text{-}ATK}$ security of regular KEM to $\mathsf{IND\text{-}ATK}^{\mathsf{XR}}$ for $\mathsf{ATK} = \{\mathsf{CPA}, \mathsf{CCA}\}$. Roughly speaking, we say an XR-KEM is secure if the hints generated by $\mathsf{HintGen}$ would not help the adversary to break the security of the challenge ciphertext.

Specifically, let KEM be an XR-KEM and we provide the security game of KEM in Figure 15. With the game $\mathsf{Game}^{\mathsf{IND\text{-}ATK}^{\mathsf{XR}}}_{\mathsf{KEM},N,b,\mathcal{A}}(\lambda)$, we say KEM is $\mathsf{IND\text{-}ATK}^{\mathsf{XR}}$ secure if for all PPT adversary $\mathcal{A}$, the following advantage $\mathsf{Adv}^{\mathsf{IND\text{-}ATK}^{\mathsf{XR}}}_{\mathsf{KEM},N,\mathcal{A}}(\lambda)$ is negligible with λ,

$$\left|\Pr\left[\mathsf{GAME}^{\mathsf{IND\text{-}ATK}^{\mathsf{XR}}}_{\mathsf{KEM},N,0,\mathcal{A}}(\lambda) = 0\right] - \Pr\left[\mathsf{GAME}^{\mathsf{IND\text{-}ATK}^{\mathsf{XR}}}_{\mathsf{KEM},N,1,\mathcal{A}}(\lambda) = 0\right]\right|.$$

B.1 Generic Construction of mmKEM from XR-KEM

Construction B.2 (XR-KEM→mmKEM Compiler). For $\mathsf{ATK} \in \{\mathsf{CPA}, \mathsf{CCA}\}$, let $\mathsf{KEM} = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Encap} = (\mathsf{Enc}^\mathsf{i}, \mathsf{Encap}^\mathsf{d}), \mathsf{Decap})$ be a (decomposable) extended reproducible $\mathsf{IND\text{-}ATK}^{\mathsf{XR}}$ secure KEM with public-private key space $\mathcal{K}$ and independent-dependent randomness distributions $(\mathcal{D}_\mathsf{i}, \mathcal{D}_\mathsf{d})$. Let $\mathsf{Compress}$, $\mathsf{Decompress}$ be the compression and decompression algorithms which can be

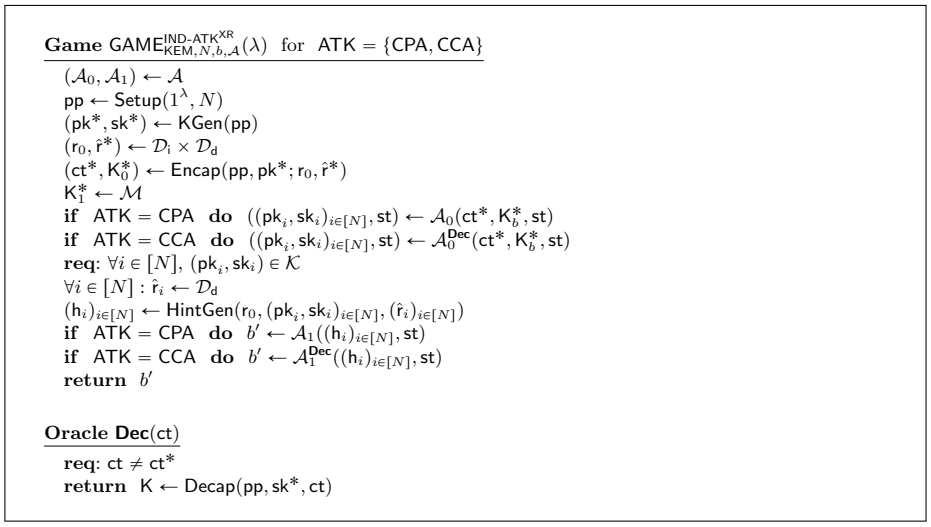

Fig. 15: The $\mathsf{IND\text{-}ATK^{XR}}$ security game for KEM with $\mathsf{ATK} = \{\mathsf{CPA}, \mathsf{CCA}\}$.

ignored if there does not exist suitable algorithms. The constructions of compiler $\mathsf{Comp}^{\mathsf{mmKEM}}[\mathsf{KEM}]$ is defined in Figure 16, which outputs an $\mathsf{mmIND\text{-}ATK^{KOSK}}$ secure mmKEM.

The correctness and security of mmKEM compiler are analogous to the mmPKE compiler in Construction 3.4. Briefly speaking, our Construction B.2 correct if the input KEM is correct as well as the output of decompression algorithm $\mathsf{Decompress}$ can still be successfully decapsulated overwhelmingly. The reduction of our Construction B.2 is the same as the one in Construction 3.4 except that we replace the message by the encapsulation key.

Appendix C Implementation and Benchmarking Details

In this section, we detail the implementation aspects of our CPA-secure primitives $\mathsf{mmCipher\text{-}KEM}$ (Cons. B.2+4.8) and $\mathsf{mmCipher\text{-}PKE}$ (Cons. 3.4+4.4) and provide further benchmarks.

We list some key technical similarities and differences between Kyber (ML-KEM) and $\mathsf{mmCipher}$ that impact performance characteristics. In the following, references are made to Kyber components as described in the ML-KEM standard FIPS 203 [54] rather than in the original Kyber paper [14]. Our comparison uses the reference code and parameter sets of the final standard.

- The programming interfaces of $\mathsf{mmCipher}$ are designed for batch encryption of unique messages/shared secrets to a large number of recipients. This is the main use case and optimization target of the implementation.
- The parameter selection of the implementation supports 2^{10} recipients, which requires a larger modulus $q = 2^{25} - 2^{12} + 1$. Hence, the Number Theoretic

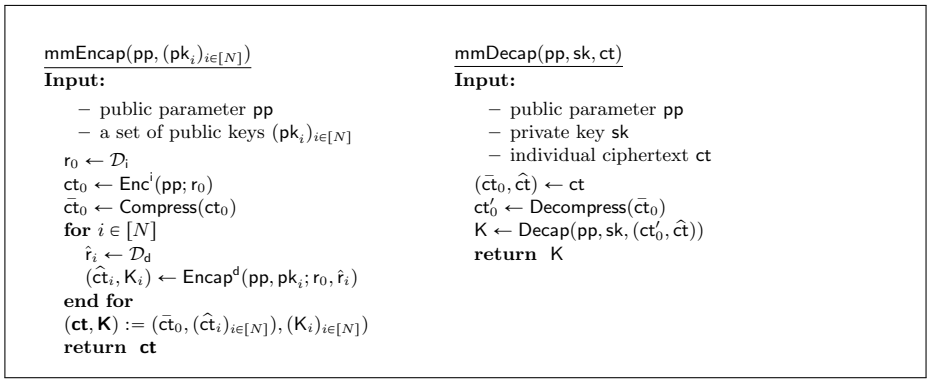

Fig. 16: The mmIND-ATK$^{\mathsf{KOSK}}$ mmKEM output by the compiler Comp$^{\mathsf{mmKEM}}$[KEM] for ATK $\in \{\mathsf{CPA}, \mathsf{CCA}\}$, where mmSetup, mmKGen, mmExt are the same as the ones in Construction 3.4.

Transforms (NTTs) operate on 32-bit elements rather than 16-bit elements, as with Kyber's $q = 3329$. Our 25-bit ring is quite similar to the 23-bit, degree-256 ring of Dilithium (ML-DSA [53]). The binary and ternary secret distributions of mmCipher also allow efficient non-NTT ring multiplication operations based on conditional additions, although these may be difficult to implement in constant time in software.

– We use the SHAKE eXtendable-Output Function (XOF) [52] for all random sampling, as is done in ML-KEM. SHAKE-128 is used for all operations at the 128-bit security level and for $\mathbf{A}$ matrix expansion at all security levels (as in ML-KEM). SHAKE-256 is used for other samplers and hashes at levels 192 and 256.
– Secret keys are sampled from a narrow uniform distribution $\mathcal{U}(\mathbb{S}_\nu)$ instead of a Centered Binomial Distribution (CBD) as in ML-KEM. The ternary ($\bar{\nu} = 3$) sampler uses rejection sampling of bytes against $3^5 = 243$; only $1 - 243/256 \approx 5\%$ of bytes are rejected, while accepted bytes yield 5 ternary digits $\{-1, 0, +1\}^5$. The "base-243" system also allows a convenient and compact storage format for ternary secret keys. Binary ($\bar{\nu} = 2$) secret key sampling and storage is trivial and optimally efficient.
– We sample ephemeral randomness from discrete Gaussian distributions $\mathcal{D}_{\sigma_0}$ and $\mathcal{D}_{\sigma_1}$ rather than from CBD. [9] Note that Gaussian widths σ are related to standard deviation $\mathfrak{s}$ by $\mathfrak{s} = \sigma/\sqrt{2\pi}$. More precisely, following Section 4.5, we fix the Gaussian width $\sigma_0 = 15.90$ and $\sigma_1 = 368459.34, 488797.36, 554941.07$

[9] Current artifact code uses rounded Gaussians for $\mathcal{D}_{\sigma_0}$ and $\mathcal{D}_{\sigma_1}$, using the polar Marsaglia method implemented with 64-bit IEEE 754 arithmetic to sample from a rounding-compensated $s' = \sqrt{\sigma^2/2\pi - 1/12}$ continuous Gaussian distribution. This sampler is approximate and not constant-time; it is a placeholder implementation. A more appropriate Discrete Gaussian sampler is required for production-level implementation.

to support up to 2^{10} recipients at 128-bit, 192-bit, and 256-bit security, respectively.

- The encryption/decryption mechanism of mmCipher-PKE is similar to Kyber, but mmCipher-KEM uses a reconciliation mechanism over $\mathcal{R}_q$, requiring the cross-rounding function $\langle\cdot\rangle_2$ and the reconciliation function $\mathsf{rec}(\cdot,\cdot)$. The Python implementation also has the randomized doubling function $\mathsf{dbl}(\cdot)$ available, but since entropy leakage ("bias") fixed by $\mathsf{dbl}(\cdot)$ can be shown to be practically negligible with our q value, the C code does not implement randomization here. These implementations are interoperable (and produce fully matching ciphertext with high probability.)
- Since SHAKE/SHA3 [52] computation is typically the biggest individual ML-KEM performance bottleneck (on some platforms consuming more than half of total key establishment cycles), for a fair comparison, the underlying KECCAK-$p[1600, 24]$ permutation implementation in mmCipher is the same plain C code as in the Kyber reference implementation.

Note that these algorithms would greatly benefit from hand-crafted SIMD and vectorization optimizations (e.g., AVX-512 or ARM SVE2). However, we currently only have a portable C implementation for mmCipher, so we are comparing such implementations of both schemes.

Table 6 includes more comprehensive benchmark results, including cycle counts for mmCipher-KEM encapsulation, mmCipher-PKE encryption, and K-PKE.Encrypt() of ML-KEM (Kyber) with various N levels up to $N = 1024$. The bandwidth of all the operations is presented in Tables 7 to 9.

Table 6: Per-message/key encryption or encapsulation latency in cycles (batch timing divided by the number of recipients N.) Note that ML-KEM becomes slower with larger N due to cache effects, whereas mmCipher significantly benefits from batching.

Scheme	$N = 2^0$	$N = 2^2$	$N = 2^4$	$N = 2^6$	$N = 2^8$	$N = 2^9$	$N = 2^{10}$
mmCipher-PKE-128	270,208	97,295	54,410	43,800	42,819	42,447	42,342
mmCipher-KEM-128	268,764	94,633	52,899	42,309	41,380	41,259	41,120
ML-KEM-512	111,006	110,929	111,025	111,301	117,662	117,679	117,665
mmCipher-PKE-192	509,848	167,253	81,055	58,540	54,873	54,126	53,849
mmCipher-KEM-192	512,542	164,080	79,468	57,706	53,774	53,265	53,138
ML-KEM-768	177,324	177,528	177,353	191,047	192,014	191,776	191,794
mmCipher-PKE-256	660,108	205,586	93,958	65,735	60,495	59,470	58,798
mmCipher-KEM-256	647,983	199,270	89,458	61,087	56,040	54,920	54,458
ML-KEM-1024	260,478	260,322	260,817	286,016	285,937	285,846	285,729

Table 7: Bandwidth of multi-recipient ciphertext $|\mathsf{ct}|$, individual ciphertext $|\mathsf{ct}_i|$, and total public keys $|\mathsf{pk}|$ for N recipients, aiming at 128-bit security.

Recipt.	Multi. Cipher. $\|\mathsf{ct}\|$ (KB)			Indiv. Cipher. $\|\mathsf{ct}_i\|$ (KB)			Total Public Keys $\|\mathsf{pk}\|$ (KB)		
N	ML KEM	mmCipher KEM	mmCipher PKE	ML KEM	mmCipher KEM	mmCipher PKE	ML KEM	mmCipher KEM	mmCipher PKE
1	0.75	1.28	1.31	0.75	1.28	1.31	0.78	3.16	3.16
16	12.00	1.75	2.25	0.75	1.28	1.31	12.50	50.50	50.50
64	48.00	3.25	5.25	0.75	1.28	1.31	50.00	202.00	202.00
256	192.00	9.25	17.25	0.75	1.28	1.31	200.00	808.00	808.00
512	384.00	17.20	33.25	0.75	1.28	1.31	400.00	1616.00	1616.00
1024	768.00	33.25	65.25	0.75	1.28	1.31	800.00	3232.00	3232.00

Table 8: Bandwidth of multi-recipient ciphertext $|\mathsf{ct}|$, individual ciphertext $|\mathsf{ct}_i|$, and total public keys $|\mathsf{pk}|$ for N recipients, aiming at 192-bit security.

Recipt.	Multi. Cipher. $\|\mathsf{ct}\|$ (KB)			Indiv. Cipher. $\|\mathsf{ct}_i\|$ (KB)			Total Public Keys $\|\mathsf{pk}\|$ (KB)		
N	ML KEM	mmCipher KEM	mmCipher PKE	ML KEM	mmCipher KEM	mmCipher PKE	ML KEM	mmCipher KEM	mmCipher PKE
1	1.06	2.44	2.47	1.06	2.44	2.47	1.16	5.50	5.50
16	17.00	2.91	3.41	1.06	2.44	2.47	18.56	88.00	88.00
64	68.00	4.41	6.41	1.06	2.44	2.47	74.24	352.00	352.00
256	272.00	10.41	18.41	1.06	2.44	2.47	296.96	1408.00	1408.00
512	544.00	18.41	34.41	1.06	2.44	2.47	593.92	2816.00	2816.00
1024	1088.00	34.41	66.41	1.06	2.44	2.47	1187.84	5632.00	5632.00

Table 9: Bandwidth of multi-recipient ciphertext $|\mathsf{ct}|$, individual ciphertext $|\mathsf{ct}_i|$, and total public keys $|\mathsf{pk}|$ for N recipients, aiming at 256-bit security.

Recipt.	Multi. Cipher. $\|\mathsf{ct}\|$ (KB)			Indiv. Cipher. $\|\mathsf{ct}_i\|$ (KB)			Total Public Keys $\|\mathsf{pk}\|$ (KB)		
N	ML KEM	mmCipher KEM	mmCipher PKE	ML KEM	mmCipher KEM	mmCipher PKE	ML KEM	mmCipher KEM	mmCipher PKE
1	1.53	3.13	3.16	1.53	3.13	3.16	1.53	7.06	7.06
16	24.50	3.59	4.09	1.53	3.13	3.16	24.48	112.96	112.96
64	98.00	5.09	7.09	1.53	3.13	3.16	97.92	451.84	451.84
256	392.00	11.09	19.09	1.53	3.13	3.16	391.68	1807.36	1807.36
512	784.00	19.09	35.09	1.53	3.13	3.16	783.36	3614.72	3614.72
1024	1568.00	35.09	67.09	1.53	3.13	3.16	1566.72	7229.44	7229.44

C.1 Additional Parameter Settings

We present the additional parameter settings with their resulting bandwidth costs in Tables 10 to 12.

Table 10: Parameter setting and its deriving bandwidth costs of our mmCipher-KEM and mmCipher-PKE, aiming at 128-bit security level, where $d = 256$, $\nu = 1$, $\bar{\nu} = 3$, $\zeta \leqslant 2^{-128}$. Sizes are in kilo-bytes.

N	$\log q$	mn	(d_u, d_v)	(σ_0, σ_1)	$\lvert\mathsf{pp}\rvert$	$\lvert\mathsf{pk}_i\rvert$	mmCipher-KEM-128					mmCipher-PKE-128				
							$\lvert\widehat{\mathsf{ct}}\rvert$	$\lvert\widehat{\mathsf{ct}}_i\rvert$	$\lvert\mathsf{ct}_i\rvert$	$\lvert\mathbf{ct}\rvert$	$k_{\mathsf{com}}^{\mathsf{KEM}}$	$\lvert\widehat{\mathsf{ct}}\rvert$	$\lvert\widehat{\mathsf{ct}}_i\rvert$	$\lvert\mathsf{ct}_i\rvert$	$\lvert\mathbf{ct}\rvert$	$k_{\mathsf{com}}^{\mathsf{PKE}}$
2^4	22	4 4	$(10, 2)$	$(15.9, 46057)$	11.00	2.75	1.25	0.03	1.28	1.75	6.86	1.25	0.06	1.31	2.25	5.33
2^7	23	4 4	$(10, 2)$	$(15.9, 130270)$	11.50	2.88	1.25	0.03	1.28	5.25	18.29	1.25	0.06	1.31	9.25	10.38
2^{10}	25	4 4	$(10, 2)$	$(15.9, 368459)$	12.50	3.13	1.25	0.03	1.28	33.25	23.10	1.25	0.06	1.31	65.25	11.77
2^{15}	27	5 5	$(11, 2)$	$(15.9, 2332958)$	21.00	4.22	1.71	0.03	1.75	1026	23.96	1.71	0.06	1.78	2050	11.99

Table 11: Parameter setting and its deriving bandwidth costs of our mmCipher-KEM and mmCipher-PKE, aiming at 192-bit security level, where $d = 256$, $\nu = 1$, $\bar{\nu} = 2$, $\zeta = 2^{-128}$. Sizes are in kilo-bytes.

N	$\log q$	mn	(d_u, d_v)	(σ_0, σ_1)	$\lvert\mathsf{pp}\rvert$	$\lvert\mathsf{pk}_i\rvert$	mmCipher-KEM-192					mmCipher-PKE-192				
							$\lvert\widehat{\mathsf{ct}}\rvert$	$\lvert\widehat{\mathsf{ct}}_i\rvert$	$\lvert\mathsf{ct}_i\rvert$	$\lvert\mathbf{ct}\rvert$	$k_{\mathsf{com}}^{\mathsf{KEM}}$	$\lvert\widehat{\mathsf{ct}}\rvert$	$\lvert\widehat{\mathsf{ct}}_i\rvert$	$\lvert\mathsf{ct}_i\rvert$	$\lvert\mathbf{ct}\rvert$	$k_{\mathsf{com}}^{\mathsf{PKE}}$
2^4	23	7 7	$(10, 2)$	$(15.9, 61099)$	35.22	5.03	2.19	0.03	2.22	2.69	6.33	2.19	0.06	2.25	3.19	5.33
2^7	24	7 7	$(10, 2)$	$(15.9, 172815)$	36.75	5.25	2.19	0.03	2.22	6.19	21.98	2.19	0.06	2.25	10.19	13.35
2^{10}	25	7 7	$(11, 2)$	$(15.9, 488797)$	38.28	5.47	2.41	0.03	2.44	34.41	31.62	2.41	0.06	2.47	66.41	16.38
2^{15}	28	8 8	$(11, 2)$	$(15.9, 2957944)$	56.00	7.00	2.75	0.03	2.78	1027	33.91	2.75	0.06	2.81	2051	16.98

Table 12: Parameter setting and its deriving bandwidth costs of our mmCipher-KEM and mmCipher-PKE, aiming at 256-bit security level, where $d = 256$, $\nu = 1$, $\bar{\nu} = 2$, $\zeta = 2^{-128}$. Sizes are in kilo-bytes.

N	$\log q$	mn	(d_u, d_v)	(σ_0, σ_1)	$\lvert\mathsf{pp}\rvert$	$\lvert\mathsf{pk}_i\rvert$	mmCipher-KEM-256					mmCipher-PKE-256				
							$\lvert\widehat{\mathsf{ct}}\rvert$	$\lvert\widehat{\mathsf{ct}}_i\rvert$	$\lvert\mathsf{ct}_i\rvert$	$\lvert\mathbf{ct}\rvert$	$k_{\mathsf{com}}^{\mathsf{KEM}}$	$\lvert\widehat{\mathsf{ct}}\rvert$	$\lvert\widehat{\mathsf{ct}}_i\rvert$	$\lvert\mathsf{ct}_i\rvert$	$\lvert\mathbf{ct}\rvert$	$k_{\mathsf{com}}^{\mathsf{PKE}}$
2^4	23	8 8	$(11, 2)$	$(15.9, 65361)$	46.00	5.75	2.75	0.03	2.78	3.25	7.54	2.75	0.06	2.81	3.75	6.53
2^7	24	9 9	$(11, 2)$	$(15.9, 196201)$	60.75	6.75	3.09	0.03	3.13	7.09	27.64	3.09	0.06	3.16	11.09	17.67
2^{10}	25	9 9	$(11, 2)$	$(15.9, 554941)$	63.28	7.03	3.09	0.03	3.13	35.09	44.69	3.09	0.06	3.16	67.09	23.37
2^{15}	28	1010	$(11, 2)$	$(15.9, 3310769)$	87.50	8.75	3.44	0.03	3.47	1027	48.84	3.44	0.06	3.50	2051	24.46

Appendix D Our Adaptively Secure mmPKE

In this section, we propose a generic construction that transforms a CPA-secure mmPKE into an adaptively secure mmPKE. Our approach generalizes the Naor-Yung paradigm [51, 61] to mmPKE, introducing an optimization: we merge the double encryption into a single multi-recipient ciphertext, only need to generate a single *independent* ciphertext. This optimization significantly reduces the size of both multi-recipient and individual ciphertexts. With our construction, not only can our lattice-based mmPKEs be transformed to achieve adaptive security, but also can the traditional mmPKEs proposed in [9, 10, 43, 60].

Compared to other adaptively secure (m)PKE constructions [5, 35, 39], our approach requires only the addition of NIZK proofs. These proofs can be aggregated, making the size constant or polylogarithmic in the number of recipients, and verification can be delegated to a server, making our construction remain both flexible and efficient, especially for large numbers of recipients. Last, we provide an instantiation in Appendix E.2.

In addition, our constructions also imply an adaptive corruption compiler which enables both CPA- and CCA-secure mmPKEs, such as the ones in [9, 10, 43, 60], to resist adaptive corruption, with some requiring KOSK assumption removal through our KOSK compiler in advance.

Construction D.1 (Adaptive Security Compiler). Let mmPKE' be an $\mathsf{mmIND\text{-}CPA}$ secure mmPKE with the randomness distributions $\mathcal{D}_{\mathsf{i}}$, $\mathcal{D}_{\mathsf{d}}$. Let Π' be a NIZK argument system. Denote the relation $R_{\Pi'}$ in Π' as

$$\left\{ \begin{pmatrix} ((\mathsf{pp}', \mathsf{pk}_0, \mathsf{pk}_1, \\ \mathsf{ct}_0, \widehat{\mathsf{ct}}_0, \widehat{\mathsf{ct}}_1, \beta); \\ (\mathsf{m}, \mathsf{r}_0, \hat{\mathsf{r}}_0, \hat{\mathsf{r}}_1)) \end{pmatrix} \middle| \begin{array}{c} \mathsf{ct}_0 = \mathsf{mmPKE}'.\mathsf{mmEnc}^{\mathsf{i}}(\mathsf{pp}'; \mathsf{r}_0) \wedge \\ \widehat{\mathsf{ct}}_0 = \mathsf{mmPKE}'.\mathsf{mmEnc}^{\mathsf{d}}(\mathsf{pp}', \mathsf{pk}_\beta, \mathsf{m}; \mathsf{r}_0, \hat{\mathsf{r}}_\beta) \wedge \\ \widehat{\mathsf{ct}}_1 = \mathsf{mmPKE}'.\mathsf{mmEnc}^{\mathsf{d}}(\mathsf{pp}', \mathsf{pk}_{1-\beta}, \mathsf{m}; \mathsf{r}_0, \hat{\mathsf{r}}_{1-\beta}) \end{array} \right\}.$$

The construction of compiler $\mathsf{Comp}^{\mathsf{CCA}}[\mathsf{mmPKE}', \Pi']$ is defined in Figure 17 which outputs an $\mathsf{mmIND\text{-}CCA}^{\mathsf{Cor}}$ secure mmPKE.

The correctness is direct. We show how to reduce the security of mmPKE output by $\mathsf{Comp}^{\mathsf{CCA}}[\mathsf{mmPKE}', \Pi']$ to the input mmPKE' and Π'. The proof is provided in Appendix G.6.

Theorem D.2 (Security). *If* mmPKE' *is* $\mathsf{mmIND\text{-}CPA}$ *secure and* Π' *is a NIZK argument system satisfies correctness, zero knowledge, and simulation soundness, our* $\mathsf{mmPKE} \leftarrow \mathsf{Comp}^{\mathsf{CCA}}[\mathsf{mmPKE}', \Pi']$ *output by Construction D.1 is* $\mathsf{mmIND\text{-}CCA}^{\mathsf{Cor}}$ *secure.*

Remark D.3 (Batch Proof and Delegate Verification). In practice, the verification of π_i can be delegated to some semi-honest third party, e.g., delivery service server. In this case, the encryptor can batch (aggregate) the proof together, i.e., generating a single proof π for the statement-witness pair $((\mathsf{pp}', (\mathsf{pk}_0^{(i)}, \mathsf{pk}_1^{(i)})_{i\in[N]}, \mathsf{ct}_0, (\widehat{\mathsf{ct}}_0^{(i)}, \widehat{\mathsf{ct}}_1^{(i)})_{i\in[N]}, \vec{\beta}), ((\mathsf{m}_i)_{i\in[N]}, \mathsf{r}_0, (\hat{\mathsf{r}}_0^{(i)}, \hat{\mathsf{r}}_1^{(i)})_{i\in[N]}))$ un-

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

Input: Public parameter $\mathsf{pp} = (\mathsf{pp}', \mathsf{crs}_{\Pi'})$

for all $i \in \{0,1\}$
$(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{mmPKE}'.\mathsf{mmKGen}(\mathsf{pp}')$
end for
$\alpha \leftarrow \{0,1\}$
return $(\mathsf{pk} := (\mathsf{pk}_0, \mathsf{pk}_1),\ \mathsf{sk} := (\alpha, \mathsf{sk}_\alpha))$

$\underline{\mathsf{mmEnc}(\mathsf{pp}, (\mathsf{pk}_i)_{i\in[N]}, (\mathsf{m}_i)_{i\in[N]})}$

Input:

- public parameter $\mathsf{pp} = (\mathsf{pp}', \mathsf{crs}_{\Pi'})$
- a set of public keys $(\mathsf{pk}_i = (\mathsf{pk}_0^{(i)}, \mathsf{pk}_1^{(i)}))_{i\in[N]}$
- a set of messages $(\mathsf{m}_i)_{i\in[N]}$

$\mathsf{r}_0 \leftarrow \mathcal{D}_\mathsf{i}$, $\mathsf{ct}_0 \leftarrow \mathsf{mmPKE}'.\mathsf{mmEnc}^\mathsf{i}(\mathsf{pp}'; \mathsf{r}_0)$
$\vec{\beta} := (\beta_i)_{i\in[N]} \leftarrow \{0,1\}^N$
for all $i \in [N]$
$\hat{\mathsf{r}}_0^{(i)}, \hat{\mathsf{r}}_1^{(i)} \leftarrow \mathcal{D}_\mathsf{d}$
$\widehat{\mathsf{ct}}_0^{(i)} \leftarrow \mathsf{mmPKE}'.\mathsf{mmEnc}^\mathsf{d}(\mathsf{pp}', \mathsf{pk}_{\beta_i}^{(i)}, \mathsf{m}_i; \mathsf{r}_0, \hat{\mathsf{r}}_{\beta_i}^{(i)})$
$\widehat{\mathsf{ct}}_1^{(i)} \leftarrow \mathsf{mmPKE}'.\mathsf{mmEnc}^\mathsf{d}(\mathsf{pp}', \mathsf{pk}_{1-\beta_i}^{(i)}, \mathsf{m}_i; \mathsf{r}_0, \hat{\mathsf{r}}_{1-\beta_i}^{(i)})$
$\pi_i \leftarrow \Pi'.\mathsf{Prove}(\mathsf{crs}_{\Pi'}, (\mathsf{pp}', \mathsf{pk}_0^{(i)}, \mathsf{pk}_1^{(i)}, \mathsf{ct}_0, \widehat{\mathsf{ct}}_0^{(i)}, \widehat{\mathsf{ct}}_1^{(i)}, \beta_i), (\mathsf{m}_i, \mathsf{r}_0, \hat{\mathsf{r}}_0^{(i)}, \hat{\mathsf{r}}_1^{(i)}))$
end for
return $\mathbf{ct} := (\mathsf{ct}_0, (\widehat{\mathsf{ct}}_0^{(i)}, \widehat{\mathsf{ct}}_1^{(i)})_{i\in[N]}, \vec{\beta}, (\pi_i)_{i\in[N]})$

$\underline{\mathsf{mmDec}(\mathsf{pp}, \mathsf{sk}, \mathsf{ct}, \mathsf{aux})}$

Input:

- public parameter $\mathsf{pp} = (\mathsf{pp}', \mathsf{crs}_{\Pi'})$
- private key $\mathsf{sk} = (\alpha, \mathsf{sk}_\alpha)$
- ciphertext $\mathsf{ct} = (\mathsf{ct}_0, \widehat{\mathsf{ct}}_0, \widehat{\mathsf{ct}}_1, \beta, \pi)$
- auxiliary information $\mathsf{aux} := \mathsf{pk} = (\mathsf{pk}_0, \mathsf{pk}_1)$

req: $\Pi'.\mathsf{Verify}(\mathsf{crs}_{\Pi'}, (\mathsf{pp}', \mathsf{pk}_0, \mathsf{pk}_1, \mathsf{ct}_0, \widehat{\mathsf{ct}}_0, \widehat{\mathsf{ct}}_1, \beta), \pi) = 1$
return $\mathsf{m} \leftarrow \mathsf{mmPKE}'.\mathsf{mmDec}(\mathsf{pp}', (\mathsf{ct}_0, \widehat{\mathsf{ct}}_{\alpha\oplus\beta}), \mathsf{sk}_\alpha)$

Fig. 17: The adaptively secure mmPKE output by the compiler $\mathsf{Comp}^{\mathsf{CCA}}[\mathsf{mmPKE}', \Pi']$. mmExt with input index i is defined by picking the relevant components $(\mathsf{ct}_0, \widehat{\mathsf{ct}}_0^{(i)}, \widehat{\mathsf{ct}}_1^{(i)}, \beta_i, \pi_i)$ from $\mathbf{ct}$. $\mathsf{mmSetup}$ is the same as the one in Construction 4.10 except for replacing Π by Π'.

der the following relation,

$$\bar{R}_{\Pi'} := \left\{ \begin{array}{c} \mathsf{ct}_0 = \mathsf{mmPKE}'.\mathsf{mmEnc}^\mathsf{i}(\mathsf{pp}'; \mathsf{r}_0) \ \wedge \\ \forall i \in [N] : \\ \widehat{\mathsf{ct}}_0^{(i)} = \mathsf{mmPKE}'.\mathsf{mmEnc}^\mathsf{d}(\mathsf{pp}', \mathsf{pk}_{\beta_i}^{(i)}, \mathsf{m}_i; \mathsf{r}_0, \hat{\mathsf{r}}_{\beta_i}^{(i)}) \wedge \\ \widehat{\mathsf{ct}}_1^{(i)} = \mathsf{mmPKE}'.\mathsf{mmEnc}^\mathsf{d}(\mathsf{pp}', \mathsf{pk}_{1-\beta_i}^{(i)}, \mathsf{m}_i; \mathsf{r}_0, \hat{\mathsf{r}}_{1-\beta_i}^{(i)}) \end{array} \right\}.$$

Therefore, each recipient does not need to download and verify the proof during the decryption.

Remark D.4 (Adaptive Corruption Compiler). By removing the NIZK component from our CCA compiler, we obtain an adaptive corruption compiler that generalizes the double encryption technique [33, 40] to the mmPKE setting.

Appendix E NIZK Instantiations in mmPKE

In this section, we discuss the instantiations of NIZK in the generic constructions of mmPKE described in Section 4.4 and Appendix D. Since we aim for quantum security in this paper, we mainly consider the post-quantum instantiations in the (quantum) random oracle model. Briefly, there are two kinds of NIZK involved: the first is a multi-proof extractable NIZK used in the KOSK compiler (Construction 4.10), and the second is a simulation-sound NIZK used in the adaptively secure compiler (Construction D.1). Finally, we present *proof-of-concept* implementations of the NIZK instantiations, which help estimate their practical cost.

E.1 NIZK in KOSK Compiler

For the NIZK in KOSK compiler, we recommend Schnorr-like lattice-based protocols that satisfy knowledge soundness and can efficiently prove the well-formedness of ciphertexts and keys. To achieve the multi-proof extractability, we can apply Katsumata Transform [38] as demonstrated in [13, 59], which leverages an *extractable linear homomorphic commitment* (LHC) that can be seen as a linear homomorphic encryption scheme with pseudo-random public keys.

Among them, LNP22 [46] is one of the most efficient lattice-based NIZKs and has recently been implemented in the LaZer library [47]. Recent work [13] extends LNP22 to achieve multi-proof extractability, but they do not provide the implementation of this variant. Therefore, we report on the results of the regular LNP22 implementation from the LaZer library as a *proof-of-concept*.

Specifically, we need to generate the "exact" range proof for the private key $(\mathbf{s}_i, \mathbf{e}_i)$, i.e., $||(\mathbf{s}_i, \mathbf{e}_i)||_\infty \leqslant 1$, along with a linear relation $\mathbf{A}^\top \mathbf{s}_i + \mathbf{e}_i = \mathbf{b}_i$. For $\bar{\nu} = 2$ (i.e., each coefficient is in $\{0, 1\}$), we simply use the concatenation $(\ \mathbf{s}^\top \ ||\ \mathbf{e}^\top\ )$ as a binary witness, proving that $\big(\ \mathbf{A}^\top \ ||\ \mathbf{I}\ \big)(\ \mathbf{s}^\top \ ||\ \mathbf{e}^\top\ )^\top - \mathbf{b} = 0$. For ternary secrets ($\bar{\nu} = 3$), the secret key is split into binary components representing positive and negative coefficients and the proof is of the form

$$\big(\ \mathbf{A}^\top \ ||\ \mathbf{I}\ ||\ -\mathbf{A}^\top \ ||\ -\mathbf{I}\ \big)(\ \mathbf{s}_+^\top \ ||\ \mathbf{e}_+^\top \ ||\ \mathbf{s}_-^\top \ ||\ \mathbf{e}_-^\top\ )^\top - \mathbf{b} = 0.$$

During the proof, we need to first prove the witness with binary coefficients and then prove the linear relation. Here although the modulus q may be smaller than the modulus in the proof system, LNP22 and its implementation in LaZer can still prove such relations efficiently. More details can be referred their papers [46, 47].

Table 13 offers representative numbers (timings on an AMD Ryzen 7 7840U laptop, 3.3 GHz). Note that the proofs have not been optimized for size or tuned for the target security level. We observe that these NIZK proofs, which need to be verified *only once* after generation, are less than 30 KB in size. Furthermore,

proof generation and verification are very efficient. In practice, this process can be delegated to a semi-honest third party, e.g., a server, and completed in "registration" phase. Hence, this NIZK has minimal impact on the performance of both encapsulation and decapsulation.

Table 13: mmCipher public-key NIZK proof sizes, proof generation and verification timings using LNP22.

Scheme	Dist.	Proof Size	Proof time	Verify time
mmCipher-128	$\bar{\nu} = 3$	26,473 B	0.078 s	0.040 s
mmCipher-192	$\bar{\nu} = 2$	25,261 B	0.075 s	0.042 s
mmCipher-256	$\bar{\nu} = 2$	28,022 B	0.105 s	0.056 s

E.2 NIZK in Adaptively Secure Compiler

For the NIZK in adaptive secure mmPKE compiler, we recommend post-quantum (zk)SNARKs that satisfy simulation soundness and provide succinct proofs with efficient verification.

LaBRADOR [11], as one of the most compact lattice-based SNARKs, has recently been implemented in the LaZer library [47] that supports proving multiple linear relations along with ℓ_2-norm bounds of each witness. Since it uses the Fiat-Shamir transformation [31], simulation soundness can be achieved in the random oracle model (see [30] for a detailed discussion). As stated in [11], the zero-knowledge property can also be easily extended, although the LaZer library does not implement this property. Therefore, we present the results of the standard LaBRADOR implementation available in the LaZer library as a *proof-of-concept*.

Specifically, as in Construction D.1 and Remark D.3, we need to prove the message consistency of the double encryption in mmCipher-PKE, i.e.,

$$\begin{bmatrix} \lfloor \mathbf{u} \bmod 2^{d_u} \rceil_q \\ \lfloor v_0^{(0)} \bmod 2^{d_v} \rceil_q \\ \lfloor v_0^{(1)} \bmod 2^{d_v} \rceil_q \\ \vdots \\ \lfloor v_{N-1}^{(0)} \bmod 2^{d_v} \rceil_q \\ \lfloor v_{N-1}^{(1)} \bmod 2^{d_v} \rceil_q \end{bmatrix} = \begin{bmatrix} \mathbf{A} \\ (\mathbf{b}_0^{(0)})^\top \\ (\mathbf{b}_0^{(1)})^\top \\ \vdots \\ (\mathbf{b}_{N-1}^{(0)})^\top \\ (\mathbf{b}_{N-1}^{(1)})^\top \end{bmatrix} \cdot \mathbf{r} + \begin{bmatrix} \mathbf{e}_u \\ y_0^{(0)} \\ y_0^{(1)} \\ \vdots \\ y_{N-1}^{(0)} \\ y_{N-1}^{(1)} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \lfloor \frac{q}{2} \rceil \cdot m_0 \\ \lfloor \frac{q}{2} \rceil \cdot m_0 \\ \vdots \\ \lfloor \frac{q}{2} \rceil \cdot m_{N-1} \\ \lfloor \frac{q}{2} \rceil \cdot m_{N-1} \end{bmatrix} + \begin{bmatrix} \mathbf{e} \\ e_0^{(0)} \\ e_1^{(1)} \\ \vdots \\ e_{N-1}^{(0)} \\ e_{N-1}^{(1)} \end{bmatrix}$$

where $\|(\mathbf{r}, \mathbf{e}_u)\|_\infty \leqslant \beta_0$, $\|(y_0^{(0)}||y_0^{(1)}|| \ldots ||y_{N-1}^{(0)}||y_{N-1}^{(1)})\|_\infty \leqslant \beta_1$, $\|\mathbf{e}\|_\infty \leqslant \lfloor q/2^{d_u+1} \rceil$ and $\|(e_0^{(0)}||e_0^{(1)}|| \ldots ||e_{N-1}^{(0)}||e_{N-1}^{(1)})\|_\infty \leqslant \lfloor q/2^{d_v+1} \rceil$. Here $\mathbf{e}$, $e_i^{(0)}$, $e_i^{(1)}$ for $i \in [N]$ are the rounding errors.

Then, we transform the relation by subtracting $c_i^{(0)}$ and $c_i^{(1)}$ to remove the message term and combining the randomness $y_i^{(0)}$, $y_i^{(1)}$, $\mathbf{e}_u$ with the rounding error together, as follows,

$$\begin{bmatrix} \lfloor \mathbf{u} \rceil_q \\ \lfloor v_0^{(0)} \rceil_q - \lfloor v_0^{(1)} \rceil_q \\ \vdots \\ \lfloor v_{N-1}^{(0)} \rceil_q - \lfloor v_{N-1}^{(1)} \rceil_q \end{bmatrix} = \begin{bmatrix} \mathbf{A} \\ (\mathbf{b}_0^{(0)})^\top - (\mathbf{b}_0^{(1)})^\top \\ \vdots \\ (\mathbf{b}_{N-1}^{(0)})^\top - (\mathbf{b}_{N-1}^{(1)})^\top \end{bmatrix} \cdot \mathbf{r} + \begin{bmatrix} \bar{\mathbf{e}} \\ \bar{e}_0 \\ \vdots \\ \bar{e}_{N-1} \end{bmatrix} \tag{E.1}$$

where $\|\mathbf{r}\|_\infty \leqslant \beta_0$, $\|\bar{\mathbf{e}}\|_\infty \leqslant \beta_0 + \lfloor q/2^{d_u+1} \rceil$ and $\|(\bar{e}_0 || \ldots || \bar{e}_{N-1})\|_\infty \leqslant 2 \cdot \lfloor q/2^{d_v+1} \rceil + 2 \cdot \beta_1 < \lfloor q/4 \rceil$. The above relation guarantees that the decryption result of the double encryption is identical.

However, it is nontrivial to directly prove Equation (E.1) via LaBRADOR in LaZer library. We identity the following challenges:

- The LaBRADOR implementation in the LaZer library supports only four different moduli, none of which can ensure that wrap-around does not occur in Equation (E.1) when switching the modulus of mmPKE to the one in LaBRADOR.
- LaBRADOR does not support proving exact ℓ_∞-norm bound of the witness.
- The norms of the error terms $\bar{e}_i$ are significantly large and exceed the range supported by LaBRADOR.

We take the following steps to solve these issues:

- We tailor a new parameter set for mmCipher-PKE shown in Table 14 with the modulus $q = 2^{32} - 99$ that is supported in the implementation of LaBRADOR. This setting can avoid the proof for switching different modulus.
- Since LaBRADOR (implementation) only supports proving the exact ℓ_2-norm of the witness, we have to use it to prove the ℓ_∞-norm, e.g., $\|r_i \in \mathcal{R}_q\|_\infty \leqslant \|r_i\|$ which introduces a relaxation factor $\sqrt{d}$. Thus, this results in an additional 4 bits per coefficient in the compressed ciphertext, which is why we set $d_v := 6$.
- We decompose each $\bar{e}_i$ and $\bar{\mathbf{e}}$ in base 8 and prove that the ℓ_∞-norm of each decomposition element is bounded by $7\sqrt{d}$.

Table 15 offers representative numbers (timings on an Intel Core i7-11850H laptop, 2.5 GHz). Note that the proofs have not been optimized for size or tuned for the target security level as well. Compared to the naive Kyber-based double encryption approach, our adaptively secure mmPKE achieves approximately a 3–6× reduction in bandwidth across different security levels. While it is about one order of magnitude slower in computation due to the NIZK overhead, the encryption time remains below 2 ms per recipient, making the scheme acceptable and practical for real-world use. As stated in Remark D.3, the verification can be delegated to a semi-honest third party, e.g., a delivery service server. Thus, this NIZK has no impact on the recipient's decryption performance.

Table 14: Parameter set for our lattice-based adaptively secure mmPKE for 256-bit message, aiming at ζ-correctness with $\zeta \leqslant 2^{-128}$, $N = 512$ recipients (i.e., 1024 ciphertexts). Note that here $|\mathbf{ct}|$ excludes the proof size.

q	d	m	n	$(\nu, \bar{\nu})$	(d_u, d_v)	(σ_0, σ_1)	pq-sec	$\lvert\mathbf{ct}\rvert$
$2^{32} - 99$	256	5	5	$(1, 3)$	$(17, 6)$	$(15.9, 412412)$	128	194.6
$2^{32} - 99$	256	8	7	$(1, 3)$	$(17, 6)$	$(15.9, 506291)$	192	196.2
$2^{32} - 99$	256	11	9	$(1, 3)$	$(18, 6)$	$(15.9, 585545)$	256	198.2

Table 15: mmCipher double encryption NIZK proof sizes, proof generation and verification timings using LaBRADOR for $N = 512$ recipients (i.e., 1024 ciphertexts).

Scheme	Proof Size	Proof time	Verify time
mmCipher-PKE-128	60.75 KB	0.827 s	0.457 s
mmCipher-PKE-192	62.06 KB	0.952 s	0.547 s
mmCipher-PKE-256	60.13 KB	1.091 s	0.609 s

Appendix F Lattice-Based mmIBE

F.1 Definition

We generalize the definition of decomposable mPKE in [39] to mmIBE as follows. Basically, an mmIBE scheme allows a sender to encrypt a set of messages to a set of identities. Compared to mmPKE, here the master key generation and user private key extraction algorithms are run by some trusted party, e.g., private key generator (PKG).

Definition F.1 (Decomposable Multi-Message Multi-Recipient IBE). A decomposable mmIBE scheme with a user private key space $\mathcal{K}$, a message space $\mathcal{M}$, a multi-recipient ciphertext space $\mathcal{C}$, and an individual ciphertext space $\mathcal{C}_{\mathsf{s}}$ consists of the following algorithms. The algorithms of mmEnc, mmExt, mmDec are the same as the ones in Definition 2.5 except that we replace the public key pk by identity id. Thus, we omit them to simplify.

- $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{MasterKeyGen}(1^\lambda)$: On input a security parameter 1^λ, it outputs a public parameter pp and a master secret key msk.
- $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{Extract}(\mathsf{pp}, \mathsf{msk}, \mathsf{id})$: On input a public parameter pp, a master secret key msk, and an identity id, it outputs a private key $\mathsf{sk}_{\mathsf{id}} \in \mathcal{K}$ for identity id.

Correctness. Let $\zeta : \mathbb{N} \to [0, 1]$. We say an mmIBE scheme is ζ-correct, if for all $\lambda, N \in \mathbb{N}$ and $i \in [N]$, message $\mathsf{m}_i \in \mathcal{M}$, identity $\mathsf{id}_i \leftarrow \{0, 1\}^*$, the following

probability holds,

$$\Pr\left[\begin{array}{c|c} \exists i \in [N] : & (\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{MasterKeyGen}(1^\lambda); \\ \mathsf{mmDec}(\mathsf{pp}, \mathsf{sk}_{\mathsf{id}_i}, \mathsf{ct}_i) \neq \mathsf{m}_i & \forall i \in [N] : \\ & \mathsf{sk}_{\mathsf{id}_i} \leftarrow \mathsf{Extract}(\mathsf{pp}, \mathsf{msk}, \mathsf{id}_i); \\ & \mathbf{ct} \leftarrow \mathsf{mmEnc}(\mathsf{pp}, (\mathsf{id}_i)_{i\in[N]}, (\mathsf{m}_i)_{i\in[N]}); \\ & \mathsf{ct}_i \leftarrow \mathsf{mmExt}(\mathsf{pp}, i, \mathbf{ct}) \end{array}\right] \leqslant \zeta(\lambda).$$

Security. Adapting the definitions from [6,18], we present the security definition of mmIBE in an adaptive (multi-)ID setting, where the adversary can adaptively choose the challenge identities after the setup phase. Compared to mmPKE, the adversary here is additionally granted access to a key-extraction oracle, which allows it to request the private keys for arbitrary identities. To prevent a trivial win, we require that the challenge messages for any challenge identities whose private keys have been queried (extracted) must be identical.

Let mmIBE be an mmIBE scheme, let N, λ be an integer, let $\mathsf{ATK} \in \{\mathsf{CPA}, \mathsf{CCA}\}$. We define $\mathsf{IND\text{-}mmID\text{-}ATK}$ security of mmIBE in Figure 18. With $\mathsf{GAME}^{\mathsf{IND\text{-}mmID\text{-}ATK}}_{\mathsf{mmIBE},N,b,\mathcal{A}}(\lambda)$, we say mmIBE is $\mathsf{IND\text{-}mmID\text{-}ATK}$ secure if for all PPT adversary $\mathcal{A}$, the following $\mathsf{Adv}^{\mathsf{IND\text{-}mmID\text{-}ATK}}_{\mathsf{mmIBE},N,\mathcal{A}}(\lambda)$ is negligible with λ,

$$\left|\Pr[\mathsf{GAME}^{\mathsf{IND\text{-}mmID\text{-}ATK}}_{\mathsf{mmIBE},N,0,\mathcal{A}}(\lambda) = 1] - \Pr[\mathsf{GAME}^{\mathsf{IND\text{-}mmID\text{-}ATK}}_{\mathsf{mmIBE},N,1,\mathcal{A}}(\lambda) = 1]\right|.$$

We say $\mathcal{A}$ wins if the game outputs 1.

Game $\mathsf{GAME}^{\mathsf{IND\text{-}mmID\text{-}ATK}}_{\mathsf{mmIBE},N,b,\mathcal{A}}(\lambda)$, $\mathsf{ATK} = \mathsf{CCA}$

$(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2) \leftarrow \mathcal{A}$
$(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{MasterKeyGen}(1^\lambda)$
$\mathsf{Ext} \leftarrow \varnothing$
$((\mathsf{id}_i, \mathsf{m}^0_i, \mathsf{m}^1_i)_{i\in[\ell]}, (\mathsf{id}_i, \mathsf{m}_i)_{i\in[\ell:N]}, \mathsf{st}) \leftarrow \mathcal{A}_0^{\mathbf{Ext},\mathbf{Dec}_0}(\mathsf{pp})$
$\mathbf{ct} \leftarrow \mathsf{mmEnc}(\mathsf{pp}, (\mathsf{id}_i)_{i\in[N]}, (\mathsf{m}^b_i)_{i\in[\ell]}, (\mathsf{m}_i)_{i\in[\ell:N]})$
$b' \leftarrow \mathcal{A}_1^{\mathbf{Ext},\mathbf{Dec}_1}(\mathbf{ct}, \mathsf{st})$
req: $\forall i \in [\ell],\ \mathsf{m}^0_i = \mathsf{m}^1_i \vee (\mathsf{id}_i \notin \mathsf{Ext} \wedge |\mathsf{m}^0_i| = |\mathsf{m}^1_i|)$
return $[b = b']$

Oracle $\mathbf{Ext}(\mathsf{id})$

$\mathsf{Ext}+ \leftarrow \mathsf{id}$
$\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{Extract}(\mathsf{pp}, \mathsf{msk}, \mathsf{id})$
return $\mathsf{sk}_{\mathsf{id}}$

Oracle $\mathbf{Dec}_0(\mathsf{ct}, \mathsf{id})$

$\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{Extract}(\mathsf{pp}, \mathsf{msk}, \mathsf{id})$
return $\mathsf{m} \leftarrow \mathsf{mmDec}(\mathsf{pp}, \mathsf{sk}_{\mathsf{id}}, \mathsf{ct})$

Oracle $\mathbf{Dec}_1(\mathsf{ct}, \mathsf{id})$

req: $\neg(\mathsf{id} = \mathsf{id}_i \in (\mathsf{id}_j)_{j\in[\ell]} \wedge \mathsf{ct} = \mathsf{mmExt}(\mathbf{ct}, i))$
$\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{Extract}(\mathsf{pp}, \mathsf{msk}, \mathsf{id})$
return $\mathsf{m} \leftarrow \mathsf{mmDec}(\mathsf{pp}, \mathsf{sk}_{\mathsf{id}}, \mathsf{ct})$

Fig. 18: The $\mathsf{IND\text{-}mmID\text{-}ATK}$ security games for mmIBE with $\mathsf{ATK} = \mathsf{CCA}$. For $\mathsf{ATK} = \mathsf{CPA}$, the adversary $\mathcal{A}$ does not have the access of the decryption oracle $\mathbf{Dec}^0$, $\mathbf{Dec}^1$.

F.2 Construction

At a high level, following the framework of DLP IBE [25], we replace the setup and key generation algorithms in Construction 4.4 and Construction 3.4 by master key generation and user private key extraction algorithms. Here, to achieve smaller and more flexible parameters, we use the modular version of pre-image sampling algorithm in NTRU lattices [20] instead of ring version in original DLP IBE [25].

We recall the master key generation and user private key extraction algorithms in [20] that are used in our mmIBE/mmIB-KEM, shown in Figure 19. To fit our notation, we swap n and m in [20]. For more details, we refer the reader to [20].

$\underline{\mathsf{MasterKeyGen}(1^\lambda)}$
Input: security parameter 1^λ
  repeat
    Sample $\mathbf{F} \leftarrow \mathcal{D}_f$ and $\mathbf{g} \leftarrow \mathcal{D}_g$
  until $\mathbf{F}$ invertible mod q and $\|\Gamma(\mathbf{B}_{\mathbf{F},\mathbf{g}})\|_{\mathrm{GS}} \leqslant \mathrm{GS\_SLACK} \cdot q^{1/(m+1)}$
  Complete $[\mathbf{g}^\top | -\mathbf{F}]$ into a basis $\mathbf{B}_{\mathbf{F},\mathbf{g}}$ of $\mathcal{L}_{\mathrm{NTRU}}$
  Compute $\mathbf{h}^\top := \mathbf{F}^{-1}\mathbf{g} \mod q$
  Pick a hash function $\mathsf{hash} : \{0,1\}^* \to \mathcal{R}_q$
  return $(\mathsf{pp} := (\begin{bmatrix} 1 \\ \mathbf{h}^\top \end{bmatrix}, \mathsf{hash}), \mathsf{msk} := \mathbf{B}_{\mathbf{F},\mathbf{g}})$

$\underline{\mathsf{Extract}(\mathsf{pp}, \mathsf{msk}, \mathsf{id})}$
Input:
  – public parameter $\mathsf{pp} = (\begin{bmatrix} 1 \\ \mathbf{h}^\top \end{bmatrix}, \mathsf{hash})$
  – master secret key $\mathsf{msk} = \mathbf{B}_{\mathbf{F},\mathbf{g}}$
  – user identity id
  $\mu \leftarrow \mathsf{hash}(\mathsf{id}) \in \mathcal{R}_q$ and let $\mathbf{c} := (\mu, \mathbf{0}_m)$
  Compute $\mathbf{t} := \mathbf{c} \cdot \mathbf{B}_{\mathbf{F},\mathbf{g}}^{-1}$
  repeat
    Compute $\mathbf{z} \in \mathcal{R}^{m+1}$ such that $\mathbf{s} := (\mathbf{t} - \mathbf{z}) \cdot \mathbf{B}_{\mathbf{F},\mathbf{g}} \leftarrow \mathcal{D}_{\sigma,\mathcal{L}_{\mathrm{NTRU}},\mathbf{c}}$
  until $\|\mathbf{s}\|_\infty \leqslant \nu$
  Phase $(\bar{\mathbf{s}}, \bar{e}) \leftarrow \mathbf{s}$ such that $\langle \bar{\mathbf{s}}, \mathbf{h} \rangle + \bar{e} = \mu$
  return $\mathsf{sk}_{\mathsf{id}} := (\bar{\mathbf{s}}, \bar{e})$

Fig. 19: Master key generation and user private key extraction algorithms in our mmIBE/mmIB-KEM.

The correctness of our mmIBE/mmIB-KEM is reduced to the one in Construction 4.4 and Construction 4.8, respectively. The security of our mmIBE/mmIB-KEM constructions reduces to the Matrix Hint-R(M)LWE assumption, as in Construction 4.4 and Construction 4.8, and additionally relies on NTRU lattices for generating users' private keys. Notably, to answer key-extraction queries, the reduction first designates the identities with different challenge messages as its own challenge identities (public keys), and then queries its challenger to extract the private key of the other identity.

We select our parameters following the method used for mmPKE in Section 4.5, and evaluate their security against known attacks on NTRU lattices using the

scripts provided in [20]. The resulting parameter sets are listed in Table 16. To ensure a fair comparison, we adopt the same root Hermite factors (RHF) of approximately 1.0075 and 1.0044 from [25], corresponding to 80-bit and 192-bit security levels, respectively.

Table 16: Parameter set for our lattice-based CPA secure mmIBE/mmIB-KEM, aiming at ζ-correctness with $\zeta \leqslant 2^{-128}$.

pq-sec	N	$\lceil \log q \rceil$	d	m	n	(d_u, d_v)	(σ_0, σ_1)	$\lvert\mathbf{ct}^{\mathsf{KEM}}\rvert$	$\lvert\mathbf{ct}^{\mathsf{PKE}}\rvert$
80	1024	44	1024	2	1	$(35, 2)$	$(15.9, 2^{38.6})$	136.75	264.75
192	1024	46	2048	2	1	$(35, 2)$	$(15.9, 2^{40.3})$	273.50	529.50

Appendix G Deferred Proofs

G.1 Proof for Generic Construction of mmPKE

We restate the Theorem 3.5 below and provide its formal proof.

Theorem G.1 (Security). *For* $\mathsf{ATK} \in \{\mathsf{CPA}, \mathsf{CCA}\}$, *if* PKE *is* $\mathsf{IND\text{-}ATK}^{\mathsf{XR}}$ *secure and satisfies extended reproducibility, our* $\mathsf{mmPKE} \leftarrow \mathsf{Comp}^{\mathsf{mmPKE}}[\mathsf{PKE}]$ *output by Construction 3.4 is* $\mathsf{mmIND\text{-}ATK}^{\mathsf{KOSK}}$ *secure.*

Proof. The proof is based on [10, Theorem 6.2]. We first consider that the case of $\mathsf{ATK} = \mathsf{CPA}$ only and then briefly indicate how to extend the argument to the case of $\mathsf{ATK} = \mathsf{CCA}$. Let $\mathcal{A}$ be a PPT adversary against the $\mathsf{mmIND\text{-}CPA}^{\mathsf{KOSK}}$ security of mmPKE. Let $\mathcal{B}$ be the reduction that utilizes the adversary $\mathcal{A}$ to break the $\mathsf{IND\text{-}CPA}^{\mathsf{XR}}$ security of PKE. The reduction $\mathcal{B}$ is described in Figure 21 where its challenger $\mathcal{C}$ is from the $\mathsf{IND\text{-}CPA}^{\mathsf{XR}}$ security game of PKE.

Like [10], we begin by defining some hybrid games associated to $\mathcal{A}$ and mmPKE in Figure 20. We parameterize these games via an index $j \in \{0, 1, \ldots, N\}$.

Denote $P_j := \Pr[\mathsf{Hyb}_j = 0]$ as the probability that experiment Hyb_j returns 0, for $j \in \{0, 1, \ldots, N\}$. We show that

$$\mathsf{Adv}^{\mathsf{mmIND\text{-}CPA}^{\mathsf{KOSK}}}_{\mathsf{mmPKE},N,\mathcal{A}}(\lambda) = P_N - P_0 \tag{G.1}$$

as follows. One can observe that

$$\Pr[\mathsf{GAME}^{\mathsf{mmIND\text{-}CPA}^{\mathsf{KOSK}}}_{\mathsf{mmPKE},N,0,\mathcal{A}} = 0] = P_N \tag{G.2}$$

$$\Pr[\mathsf{GAME}^{\mathsf{mmIND\text{-}CPA}^{\mathsf{KOSK}}}_{\mathsf{mmPKE},N,1,\mathcal{A}} = 0] = P_0 \tag{G.3}$$

since when $j = N$, the message vector inside the challenge ciphertext is $(\mathsf{m}^0_i)_{i\in[N]}$ and when $j = 0$, the one is $(\mathsf{m}^1_i)_{i\in[N]}$. Therefore, in the adversary $\mathcal{A}$'s view,

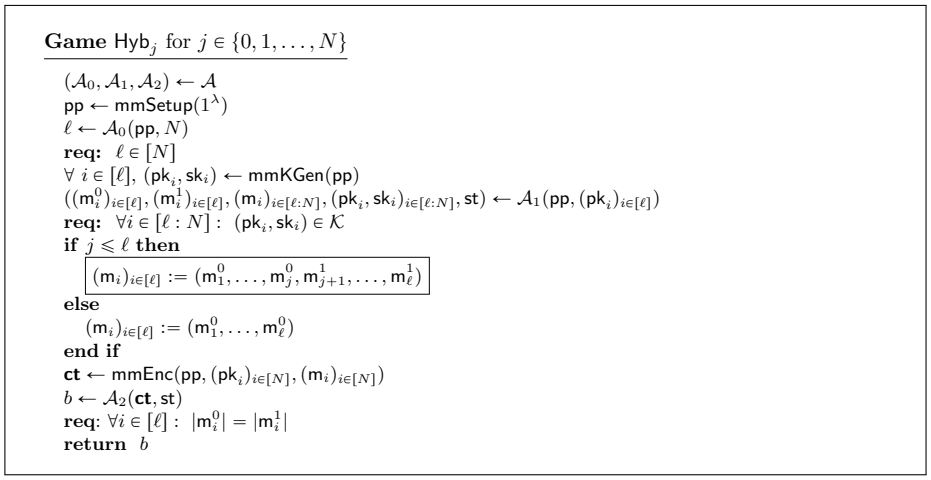

Fig. 20: The hybrid games in Theorem G.1.

the experiment Hyb_N is the same as $\mathsf{GAME}^{\mathsf{mmIND\text{-}CPA}^{\mathsf{KOSK}}}_{\mathsf{mmPKE},N,0,\mathcal{A}}$ and Hyb_0 is the same as $\mathsf{GAME}^{\mathsf{mmIND\text{-}CPA}^{\mathsf{KOSK}}}_{\mathsf{mmPKE},N,1,\mathcal{A}}$. After subtraction between Equation (G.2) and Equation (G.3), we can get Equation (G.1).

From the description of reduction $\mathcal{B}$ in Figure 21, we claim that

$$\Pr\left[\mathsf{GAME}^{\mathsf{IND\text{-}CPA}^{\mathsf{XR}}}_{\mathsf{PKE},N,0,\mathcal{B}}(\lambda) = 0\right] = \frac{1}{N} \cdot \sum_{j=1}^{N} P_j \ , \tag{G.4}$$

$$\Pr\left[\mathsf{GAME}^{\mathsf{IND\text{-}CPA}^{\mathsf{XR}}}_{\mathsf{PKE},N,1,\mathcal{B}}(\lambda) = 0\right] = \frac{1}{N} \cdot \sum_{j=1}^{N} P_{j-1} \ . \tag{G.5}$$

We explain the reason of the above equations holding as follows. Firstly, each index $j \in [N]$ is equally likely for the reduction $\mathcal{B}$ and then the j-th extracted individual ciphertext ct_j from the adversary $\mathcal{A}$'s multi-recipient challenge ciphertext $\mathbf{ct}$ is the reduction $\mathcal{B}$'s challenge ciphertext ct^*. Furthermore, due to the extended reproducibility of PKE, all extracted individual ciphertexts $(\mathsf{ct}_i)_{i\in[N]}$ from the multi-recipient challenge ciphertext $\mathbf{ct}$ are generated using the same randomness r_0 and different randomness $(\hat{\mathsf{r}}_i)_{i\in[N]}$. Therefore, one can observe that the game $\mathsf{GAME}^{\mathsf{IND\text{-}CPA}^{\mathsf{XR}}}_{\mathsf{PKE},N,0,\mathcal{B}}(\lambda)$ is the same as Hyb_j and the game $\mathsf{GAME}^{\mathsf{IND\text{-}CPA}^{\mathsf{XR}}}_{\mathsf{PKE},N,1,\mathcal{B}}(\lambda)$ is the same as Hyb_{j-1}.

Then after the subtraction between Equation (G.4) and Equation (G.5), we can obtain

$$\begin{aligned}\mathsf{Adv}^{\mathsf{IND\text{-}CPA}^{\mathsf{XR}}}_{\mathsf{PKE},N,\mathcal{B}}(\lambda) &= \Pr\left[\mathsf{GAME}^{\mathsf{IND\text{-}CPA}^{\mathsf{XR}}}_{\mathsf{PKE},N,0,\mathcal{B}}(\lambda)=0\right] - \Pr\left[\mathsf{GAME}^{\mathsf{IND\text{-}CPA}^{\mathsf{XR}}}_{\mathsf{PKE},N,1,\mathcal{B}}(\lambda)=0\right] \\ &= \frac{1}{N}\cdot\left(\sum_{j=1}^{n} P_j - \sum_{j=1}^{n} P_{j-1}\right) = \frac{1}{N}\cdot(P_N - P_0) \\ &= \frac{1}{N}\cdot \mathsf{Adv}^{\mathsf{mmIND\text{-}CPA}^{\mathsf{KOSK}}}_{\mathsf{mmPKE},N,\mathcal{A}}(\lambda).\end{aligned}$$

And the running time of the reduction $\mathcal{B}$ is the sum of the adversary $\mathcal{A}$ and the reproduce algorithm Rep. Overall, we get the security of mmPKE.

Here, we briefly discuss how to extend the above proof to the case of $\mathsf{ATK} = \mathsf{CCA}$. The definition of the hybrid games is the same as in Figure 20. We show how the reduction $\mathcal{B}$ answers the decryption queries from the adversary $\mathcal{A}$. First of all, the reduction $\mathcal{B}$ is also given the access of the decryption oracle of $\mathsf{IND\text{-}CCA}^{\mathsf{XR}}$ secure PKE. Therefore, when requiring to decrypt the individual ciphertext for the public key pk_j, $\mathcal{B}$ will provide the answer by invoking its own given decryption oracle. For the individual ciphertext for the other public key, i.e., pk_i for $i \in [\ell]/\{j\}$, $\mathcal{B}$ can decrypt the ciphertext by itself since it is in possession of the corresponding private key sk_i. □

G.2 Proofs for Matrix Hint-MLWE

We restate Lemma 4.2 and Theorem4.3 below and provide their formal proofs.

Lemma G.2. *Let $d, \ell > 0$ be integers. Let Σ_1, $\Sigma_{\mathbf{y}}$ be positive definite symmetric matrices over $\mathbb{R}^{d\times d}$ and $\mathbb{R}^{\ell\times\ell}$, respectively. Let $R \in \mathbb{Z}^{\ell\times d}$ be an integer matrix. Denote $\Sigma_0 := \left(\Sigma_1^{-1} + R^\top \Sigma_{\mathbf{y}}^{-1} R\right)^{-1}$. Then, the following two distributions over $\mathbb{Z}^{d+\ell}$ are statistically identical:*

$$\begin{aligned}&\left\{ \left(\vec{r},\vec{h}\right) \;\middle|\; \vec{r} \leftarrow \mathcal{D}_{\mathbb{Z}^d,\sqrt{\Sigma_1}},\ \vec{y} \leftarrow \mathcal{D}_{\mathbb{Z}^\ell,\sqrt{\Sigma_{\mathbf{y}}}},\ \vec{h} = R\vec{r}+\vec{y} \right\} \\ \approx &\left\{ \left(\vec{r},\vec{h}\right) \;\middle|\; \begin{array}{c} \vec{r} \leftarrow \mathcal{D}_{\mathbb{Z}^d,\sqrt{\Sigma_1}},\ \vec{y} \leftarrow \mathcal{D}_{\mathbb{Z}^\ell,\sqrt{\Sigma_{\mathbf{y}}}},\ \vec{h} = R\vec{r}+\vec{y} \\ \vec{c} = \Sigma_0 R^\top {\Sigma_{\mathbf{y}}}^{-1}\vec{h},\ \vec{r} \leftarrow \mathcal{D}_{\mathbb{Z}^d,\vec{c},\sqrt{\Sigma_0}} \end{array} \right\}.\end{aligned}$$

Proof. The proof is similar to [42, Lemma 7]. We show that two random variables have the same probability mass function. The probability that the first random variable outputs $(\vec{v},\vec{w}) \in \mathbb{Z}^d \times \mathbb{Z}^\ell$ can be written as follows:

$$\begin{aligned}&\Pr\left[\vec{r}=\vec{v},\ R\vec{r}+\vec{y}=\vec{w} \mid \vec{r} \leftarrow \mathcal{D}_{\mathbb{Z}^d,\sqrt{\Sigma_1}},\ \vec{y} \leftarrow \mathcal{D}_{\mathbb{Z}^\ell,\sqrt{\Sigma_{\mathbf{y}}}}\right] \\ =&\mathcal{D}_{\mathbb{Z}^d,\sqrt{\Sigma_1}}(\vec{v}) \cdot \mathcal{D}_{\mathbb{Z}^\ell,\sqrt{\Sigma_{\mathbf{y}}}}(\vec{w}-R\vec{v}) \\ \propto &\exp\left[-\pi\left(\vec{v}^\top\Sigma_1^{-1}\vec{v} + (\vec{w}-R\vec{v})^\top\Sigma_{\mathbf{y}}^{-1}(\vec{w}-R\vec{v})\right)\right] \\ =&\exp\left[-\pi\left((\vec{v}-\vec{c})^\top\Sigma_0^{-1}(\vec{v}-\vec{c}) - \vec{c}^\top\Sigma_0^{-1}\vec{c} + \vec{w}^\top\Sigma_{\mathbf{y}}^{-1}\vec{w}\right)\right]\end{aligned}$$

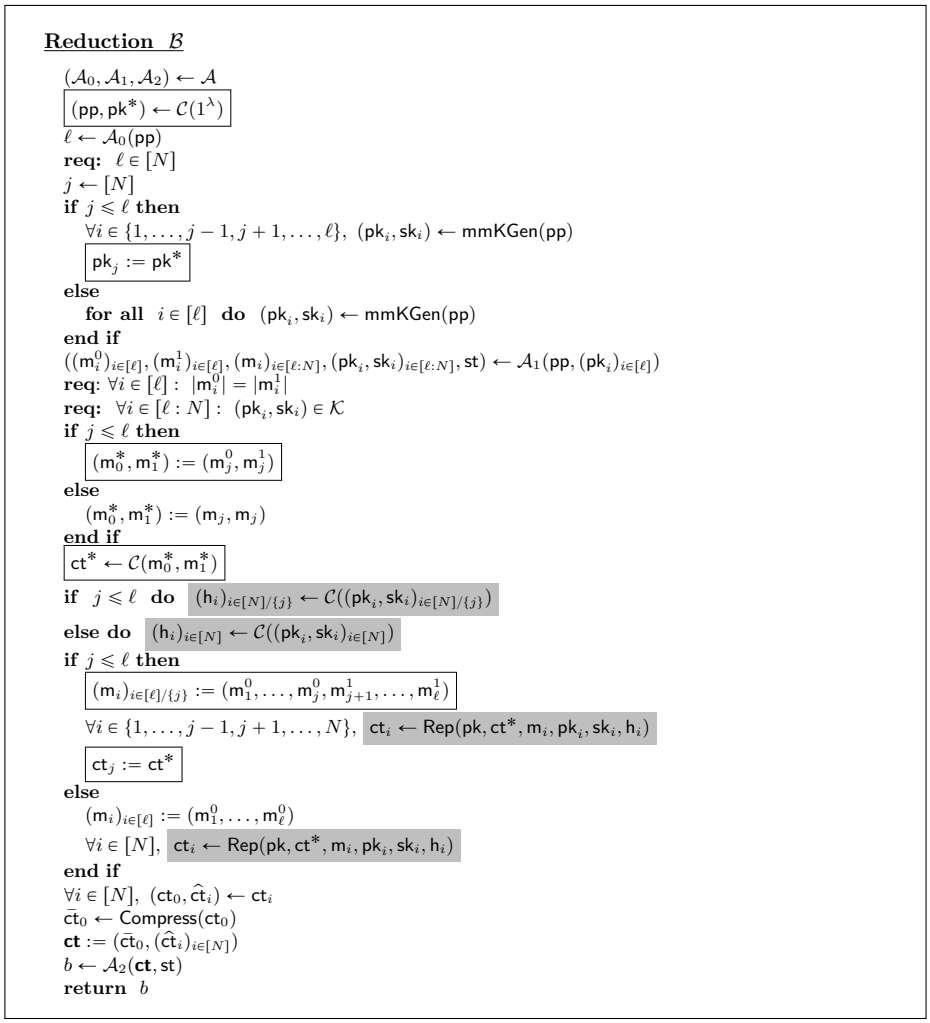

Fig. 21: The reduction $\mathcal{B}$ using the adversary $\mathcal{A}$ of mmPKE to break the security of PKE in Theorem G.1. The parts where $\mathcal{B}$'s operations are different from $\mathsf{mmIND\text{-}CPA}^{\mathsf{KOSK}}$ security game are marked by boxes. The parts which are different from the reduction in [10] are highlighted by boxes.

where $\vec{c} = \Sigma_0 R^\top \Sigma_{\mathbf{y}}^{-1} \vec{w}$.

Since the term $-\vec{c}^\top \Sigma_0^{-1} \vec{c} + \vec{w}^\top \Sigma_{\mathbf{y}}^{-1} \vec{w}$ is a constant that does not depend on $\vec{v}$ and the conditional probability $\Pr\left[\vec{r} = \vec{v} \mid R\vec{r} + \vec{y} = \vec{w}\right]$ is proportional to $\exp\left[-\pi(\vec{v} - \vec{c})^\top \Sigma_0^{-1} (\vec{v} - \vec{c})\right]$, it implies

$$\Pr\left[\vec{r} = \vec{v} \mid R\vec{r} + \vec{y} = \vec{w}\right] \equiv \rho_{\sqrt{\Sigma_0}}(\vec{v} - \vec{c}) \equiv \Pr\left[\vec{\hat{r}} = \vec{v} \mid R\vec{r} + \vec{y} = \vec{w}\right].$$

Therefore, the given two distributions are statistically identical. □

Theorem G.3 (Hardness of Matrix Hint-MLWE). *Let m, n, q, ℓ be positive integers. Let $\mathcal{S}$ be a distribution over $\mathcal{R}^{\ell\times(m+n)}$. Let $B > 0$ be a real number such that $\|\bar{R}\|^2 \leqslant B$ for any possible $\mathbf{R} \leftarrow \mathcal{S}$ and $\bar{R} := \Gamma(\mathbf{R})$. Let $\sigma_0, \sigma_1, \sigma, \delta > 0$ be real numbers. Let Σ_1, $\Sigma_{\mathbf{y}}$ be a positive definite symmetric matrices over $\mathbb{R}^{(m+n)d\times(m+n)d}$ and $\mathbb{R}^{\ell d\times\ell d}$, respectively, such that $\|\Sigma_1^{-1}\| \leqslant \frac{1}{\sigma_0^2}$ and $\|\Sigma_{\mathbf{y}}^{-1}\| \leqslant \frac{1}{\sigma_1^2}$. Let $\chi_0 := \mathcal{D}_{\mathbb{Z}^{(m+n)d},\sqrt{\Sigma_1}}$ be a distribution over $\mathcal{R}^{m+n}$, $\chi_1 := \mathcal{D}_{\mathbb{Z}^{\ell d},\sqrt{\Sigma_{\mathbf{y}}}}$ be a distribution over $\mathcal{R}^{\ell}$, and $\chi := \mathcal{D}_{\mathbb{Z}^{(m+n)d},\sigma}$ be a distribution over $\mathcal{R}^{m+n}$. There exists an efficient reduction from $\mathsf{MLWE}_{\mathcal{R},m,n,q,\chi}$ to $\mathsf{MatrixHint\text{-}MLWE}^{\ell,\chi_1,\mathcal{S}}_{\mathcal{R},m,n,q,\chi_0}$ that reduces the advantage by at most 2ϵ, if the sampleability condition*

$$\frac{1}{(1+\delta)\sigma^2+\delta_0} \geqslant \frac{1}{\sigma_0^2} + \frac{B}{\sigma_1^2} \tag{G.6}$$

where $\delta_0 := \sqrt{\frac{\ln(2(m+n)d)+4}{\pi}}$, and the convolution condition

$$\sigma \geqslant \sqrt{1+1/\delta}\cdot\eta_\varepsilon(\mathbb{Z}^{(m+n)d}) \tag{G.7}$$

are satisfied.

Specifically, for any PPT adversary $\mathcal{A}$ against the $\mathsf{MatrixHint\text{-}MLWE}^{\ell,\chi_1,\mathcal{S}}_{\mathcal{R},m,n,q,\chi_0}$ assumption, there exists a PPT adversary $\mathcal{B}$ against the $\mathsf{MLWE}_{\mathcal{R},m,n,q,\chi}$ assumption, such that

$$\mathsf{Adv}^{\mathsf{MatrixHint\text{-}MLWE}}_{\mathsf{para}_0,\mathcal{A}}(\lambda) \leqslant \mathsf{Adv}^{\mathsf{MLWE}}_{\mathsf{para}_1,\mathcal{B}}(\lambda) + 2\epsilon$$

where $\mathsf{para}_0 = ((\mathcal{R},m,n,q,\chi_0),(\ell,\chi_1,\mathcal{S}))$ and $\mathsf{para}_1 = (\mathcal{R},m,n,q,\chi)$.

Proof. The proof is based on [42, Theorem 1] and [29, Theorem 2]. With an adversary $\mathcal{A}$ against $\mathsf{MatrixHint\text{-}MLWE}^{\ell,\chi_1,\mathcal{S}}_{\mathcal{R},m,n,q,\chi_0}$, we show how the adversary $\mathcal{B}$ breaks $\mathsf{MLWE}_{\mathcal{R},m,n,q,\chi}$.

Given an $\mathsf{MLWE}_{\mathcal{R},m,n,q,\chi}$ instance $(\mathbf{A},\mathbf{b}) \in \mathcal{R}_q^{m\times n}\times\mathcal{R}_q^m$, $\mathcal{B}$ first samples $\mathbf{R}\leftarrow\mathcal{S}$, sets $\bar{R} := \Gamma(\mathbf{R})$ and

$$\Sigma_0 := \left(\Sigma_1^{-1} + \bar{R}^\top\Sigma_{\mathbf{y}}^{-1}\bar{R}\right)^{-1}.$$

Then, $\mathcal{B}$ samples the following elements over $\mathcal{R}$,

- $\mathbf{r}\leftarrow\chi_0$
- $\mathbf{y}\leftarrow\chi_1$
- $\mathbf{t}\leftarrow\mathcal{D}_{\mathbb{Z}^{(m+n)d},\vec{c},\sqrt{\Sigma_0-\sigma^2 I_{(m+n)d}}}$ where $\vec{c} = \Sigma_0\bar{R}^\top\Sigma_{\mathbf{y}}^{-1}(\bar{R}\cdot\Gamma(\mathbf{r})+\Gamma(\mathbf{y}))$

By Lemma A.1, $\mathbf{t}$ can be PPT sampled from $\mathcal{D}_{\mathbb{Z}^{(m+n)d},\vec{c},\sqrt{\Sigma_0-\sigma^2 I_{(m+n)d}}}$ if the following conditions hold: (1) Σ is positive definite where $\Sigma := \Sigma_0 - \sigma^2 I_{(m+n)d}$, i.e., $\sigma_{\min}(\Sigma) > 0$; (2) $\delta_0\cdot B_\Sigma \leqslant 1$ where $\delta_0 := \sqrt{\frac{\ln(2(m+n)d)+4}{\pi}}$ and B_Σ denotes the max value among the norm of each column of $\sqrt{\Sigma^{-1}}$. One can observe that

$$B_\Sigma \leqslant \sqrt{\sigma_{\max}(\Sigma^{-1})} \leqslant \frac{1}{\sqrt{\sigma_{\min}(\Sigma)}} = \frac{1}{\sqrt{\sigma_{\min}(\Sigma_0)-\sigma^2}}.$$

And since $\sigma_{\min}(\Sigma_0) = \frac{1}{\|\Sigma_0^{-1}\|}$, we have

$$\|\Sigma_0^{-1}\| = \|\Sigma_1^{-1} + \bar{R}^\top \Sigma_{\mathbf{y}}^{-1} \bar{R}\| \leqslant \|\Sigma_1^{-1}\| + \|\Sigma_{\mathbf{y}}^{-1}\| \cdot \|\bar{R}^\top \bar{R}\| \leqslant \frac{1}{\sigma_0^2} + \frac{B}{\sigma_1^2}$$

where the first inequality is obtained by the triangle inequality, and the second inequality uses the fact $\|\bar{R}^\top \bar{R}\| = \|\bar{R}\|^2$ and the requirement bound $\|\Sigma_1^{-1}\| \leqslant \frac{1}{\sigma_0^2}$, $\|\Sigma_{\mathbf{y}}^{-1}\| \leqslant \frac{1}{\sigma_1^2}$, $\|\bar{R}\|^2 \leqslant B$. Thus, the above two conditions for Lemma A.1 can be combined as *sampleability condition*, i.e.,

$$\sigma_{\min}(\Sigma_0) = \frac{1}{\|\Sigma_0^{-1}\|} \geqslant \frac{1}{\frac{1}{\sigma_0^2} + \frac{B^2}{\sigma_1^2}} \geqslant (1+\delta) \cdot \sigma^2 + \delta_0 \geqslant \sigma^2 \tag{G.8}$$

for some $\delta \geqslant 0$.

Later, $\mathcal{B}$ uses the sampled elements to transform the given MLWE instance $(\mathbf{A}, \mathbf{b})$ into an MatrixHint-MLWE instance and sends it to the adversary $\mathcal{A}$. Finally, $\mathcal{B}$ utilizes the reply from $\mathcal{A}$ to break MLWE. $\mathcal{B}$ starts by constructing

$$(\mathbf{A}, \mathbf{b} + [\mathbf{I}_m|\mathbf{A}]\mathbf{t}, \mathbf{R}, \mathbf{h}) \tag{G.9}$$

where $\mathbf{h} := \mathbf{R}\mathbf{r} + \mathbf{y}$.

Suppose $\mathbf{b} = [\mathbf{I}_m|\mathbf{A}]\mathbf{r}'$ where $\mathbf{r}' \leftarrow \chi$, we have

$$\mathbf{b} + [\mathbf{I}_m|\mathbf{A}]\mathbf{t} = [\mathbf{I}_m|\mathbf{A}](\mathbf{r}' + \mathbf{t})$$

where $\mathbf{r}' + \mathbf{t}$ is under the distribution

$$\mathcal{D}_{\mathbb{Z}^{(m+n)d}, \sigma \mathbf{I}_{(m+n)d}} + \mathcal{D}_{\mathbb{Z}^{(m+n)d}, \vec{c}, \sqrt{\Sigma_0 - \sigma^2 I_{(m+n)d}}}.$$

Denote $\Sigma_2^{-1} := \sigma^{-2} I_{(m+n)d} + (\Sigma_0 - \sigma^2 I_{(m+n)d})^{-1}$. By Lemma A.2, the distribution $\mathcal{D}_{\mathbb{Z}^{(m+n)d}, \sigma \mathbf{I}_{(m+n)d}} + \mathcal{D}_{\mathbb{Z}^{(m+n)d}, \vec{c}, \sqrt{\Sigma_0 - \sigma^2 I_{(m+n)d}}}$ is within the statistical distance 2ϵ of $\mathcal{D}_{\mathbb{Z}^{(m+n)d}, \vec{c}, \sqrt{\Sigma_0}}$ if $\sqrt{\Sigma_2} \geqslant \eta_\varepsilon(\mathbb{Z}^{(m+n)d})$ holds. We have $\|(\Sigma_0 - \sigma^2 I_{(m+n)d})^{-1}\| = \frac{1}{\sigma_{\min}(\Sigma_0 - \sigma^2 I_{(m+n)d})}$ and if Equation (G.8) holds, we can obtain

$$\sigma_{\min}(\Sigma_0 - \sigma^2 I_{(m+n)d}) = \sigma_{\min}(\Sigma_0) - \sigma^2 \geqslant \delta \cdot \sigma^2 + \delta_0 \geqslant \delta \cdot \sigma^2.$$

Combining the triangle inequality with Lemma A.5, we show the *convolution condition* as

$$\|\Sigma_2^{-1}\| \leqslant \frac{1}{\sigma^2} + \frac{1}{\sigma_{\min}(\Sigma_0 - \sigma^2 I_{(m+n)d})} \leqslant \frac{1 + 1/\delta}{\sigma^2} \leqslant \eta_\varepsilon(\mathbb{Z}^{(m+n)d})^{-2}. \tag{G.10}$$

If the *convolution condition* holds, the distribution of Equation (G.9) is within statistical distance 2ϵ of

$$(\mathbf{A}, [\mathbf{I}_m|\mathbf{A}]\hat{\mathbf{r}}, \mathbf{R}, \mathbf{h}) \tag{G.11}$$

where $\hat{\mathbf{r}} \leftarrow \mathcal{D}_{\mathbb{Z}^{(m+n)d}, \vec{c}, \sqrt{\Sigma_0}}$.

Then, by Lemma G.2, the distribution of $(\hat{\mathbf{r}}, \mathbf{h})$ is identical to that of $(\mathbf{r}, \mathbf{h})$. Thus, the distribution of Equation (G.11) is identical to

$$(\mathbf{A}, [\mathbf{I}_m|\mathbf{A}]\mathbf{r}, \mathbf{R}, \mathbf{h}) \tag{G.12}$$

which are the instance of $\mathsf{MatrixHint\text{-}MLWE}^{\ell,\chi_1,\mathcal{S}}_{\mathcal{R},m,n,q,\chi_0}$ assumption.

In summary, if *sampleability condition* and *convolution condition* in Equation (G.6) and (G.7) hold and the $\mathsf{MLWE}_{\mathcal{R},m,n,q,\chi}$ assumption is hard, i.e., the adversary $\mathcal{B}$ cannot distinguish between $[\mathbf{I}_m|\mathbf{A}]\mathbf{r}'$ with $\mathbf{r}' \leftarrow \chi$ and the uniformly random value $\mathbf{b} \leftarrow \mathcal{R}_q^m$, then the adversary $\mathcal{A}$ cannot distinguish between the Equation (G.9) and

$$(\mathbf{A}, \mathbf{u}, \mathbf{R}, \mathbf{h}) \tag{G.13}$$

where $\mathbf{u} \leftarrow \mathcal{R}_q^m$ is uniformly random, with additional advantage at most 2ϵ. □

G.3 Proofs for XR-PKE

We restate Theorem 4.5, Theorem 4.6, and Theorem 4.7 below and provide their formal proofs.

Theorem G.4 (Extended Reproducibility). *For any positive integer N, our* PKE *in Construction 4.4 is extended reproducible. More precisely, for the extended reproducible game in Figure 2, the following probability holds,*

$$\Pr\left[\mathsf{Game}^{\mathsf{ext\text{-}repr}}_{\mathsf{PKE},\mathsf{Rep},N}(\lambda) = 1\right] = 1.$$

Proof. Suppose $\mathsf{ct}^* := (\mathbf{c}, u^*) \leftarrow \mathsf{Enc}(\mathbf{A}, \mathbf{b}^*, m^*)$ with randomness $\mathsf{r}_0 := (\mathbf{r}, \mathbf{e}_u)$, $\hat{\mathsf{r}}^* := y^*$, where we have

$$\mathbf{c} = \mathbf{A}\mathbf{r} + \mathbf{e}_u. \tag{G.14}$$

and $u^* = \lfloor \langle \mathbf{b}^*, \mathbf{r} \rangle + y^* + \lfloor \frac{q}{2} \rceil \cdot m^* \rceil_{2^{d_v}}$. For each $i \in [N]$, the public key $\mathbf{b}_i \leftarrow \mathsf{KGen}(\mathbf{A})$. Thus, we have

$$\mathbf{b}_i = \mathbf{A}^\top \mathbf{s}_i + \mathbf{e}_i. \tag{G.15}$$

For the hints $(h_i)_{i\in[N]} \leftarrow \mathsf{HintGen}((\mathbf{r}, \mathbf{e}_u), (\mathbf{b}_i, \mathbf{e}_i)_{i\in[N]}, (y_i)_{i\in[N]})$, we have

$$h_i = \langle \mathbf{r}, \mathbf{e}_i \rangle - \langle \mathbf{e}_u, \mathbf{s}_i \rangle + y_i. \tag{G.16}$$

On input h_i, $\mathsf{Rep}((\mathbf{c}, u^*), m_i, \mathbf{b}_i, \mathbf{s}_i, h_i)$ outputs the reproduced ciphertext $(\mathbf{c}, u_i)$ for $\mathbf{b}_i$, where

$$u_i = \lfloor \langle \mathbf{c}, \mathbf{s}_i \rangle + h_i + \lfloor \frac{q}{2} \rceil \cdot m_i \rceil_{2^{d_v}}. \tag{G.17}$$

When plugging Equation (G.14), Equation (G.15), Equation (G.16) into Equation (G.17), we have

$$u_i = \lfloor \langle \mathbf{b}_i, \mathbf{r} \rangle + y_i + \lfloor \frac{q}{2} \rceil \cdot m_i \rceil_{2^{d_v}}$$

which is the same as the output from $\mathsf{Enc}(\mathbf{A}, \mathbf{b}_i, m_i; (\mathbf{r}, \mathbf{e}_u), y_i)$.

Overall, we get the extended reproducibility of our construction. □

Theorem G.5 (Correctness). *Let $\mathbf{e}, \mathbf{s}, \mathbf{r}, \mathbf{e}_u, y$ be random variables that have the corresponding distribution as in Construction 4.4. Denote ζ as*

$$\Pr\left[\ \left\|\langle \mathbf{e}, \mathbf{r}\rangle + y - \langle \mathbf{s}, \mathbf{e}_u\rangle - c_v + \langle \mathbf{s}, \mathbf{c}_u\rangle\right\|_\infty \geqslant \lfloor q/4 \rceil\ \right]$$

where $\mathbf{c}_u := \mathbf{c} - \lfloor\lfloor \mathbf{c} \bmod q\rceil_{2^{d_u}}\rceil_q \in \mathcal{R}^m$, and $c_v := c - \lfloor\lfloor c \bmod q\rceil_{2^{d_v}}\rceil_q \in \mathcal{R}$. We say our Construction 4.4 is ζ-correct.

Proof. The value u' in Dec algorithm is

$$u' := \lfloor u \mod 2^{d_v}\rceil_q = \lfloor\lfloor c \mod q\rceil_{2^{d_v}}\rceil_q .$$

Considering the compression and decompression of key-*independent* ciphertext $\mathbf{c}$, the value $\mathbf{c}$ (renamed as $\mathbf{c}'$) in Dec algorithm is

$$\mathbf{c}' := \lfloor\lfloor \mathbf{c} \mod q\rceil_{2^{d_u}}\rceil_q .$$

Plugging $\lfloor\lfloor \mathbf{c} \bmod q\rceil_{2^{d_u}}\rceil_q = \mathbf{c} - \mathbf{c}_u$, and $\lfloor\lfloor c \bmod q\rceil_{2^{d_v}}\rceil_q = c - c_v$, we have

$$u' - \langle \mathbf{c}', \mathbf{s}\rangle = c - c_v - \langle \mathbf{c} - \mathbf{c}_u, \mathbf{s}\rangle .$$

Since $c = \langle \mathbf{b}, \mathbf{r}\rangle + y + \lfloor q/2\rceil \cdot m$ and $\mathbf{c} := \mathbf{A}\mathbf{r} + \mathbf{e}_u$, where $\mathbf{b} := \mathbf{A}^\top \mathbf{s} + \mathbf{e}$, we can obtain the decryption is made by computing

$$u' - \langle \mathbf{c}', \mathbf{s}\rangle = \langle \mathbf{e}, \mathbf{r}\rangle + y - \langle \mathbf{s}, \mathbf{e}_u\rangle - c_v + \langle \mathbf{s}, \mathbf{c}_u\rangle + \lfloor q/2\rceil \cdot m .$$

It means that when ℓ_∞-norm of the decryption error is no less than $\lfloor q/4\rceil$, i.e., $\|\langle \mathbf{e}, \mathbf{r}\rangle + y - \langle \mathbf{s}, \mathbf{e}_u\rangle - c_v + \langle \mathbf{s}, \mathbf{c}_u\rangle\|_\infty \geqslant \lfloor q/4\rceil$, the decryption will fail. Thus, the probability ζ is no more than the probability of decryption failure. □

Theorem G.6 (Security). *Let m, n, d, q, N, ν be positive integers parameters. Let $\sigma, \sigma_0, \sigma_1$ be Gaussian width parameters. Let the positive real matrices Σ_1 and $\Sigma_{\mathbf{y}}$ be as Equation (4.5). Let the distribution $\mathcal{S}$ and the bound B be as Equation (4.3) and (4.4) respectively. Let the distribution $\chi_0 := \mathcal{D}_{\mathbb{Z}^{(m+n+1)d}, \sqrt{\Sigma_1}}$, $\chi_1 := \mathcal{D}_{\mathbb{Z}^{Nd}, \sqrt{\Sigma_{\mathbf{y}}}}$, $\bar{\chi} := \mathcal{U}(\mathbb{S}_\nu)$. Suppose Equation (4.1) and (4.2) hold.*

Our PKE *in Construction 4.4 is* $\mathsf{IND\text{-}CPA}^{\mathsf{XR}}$ *secure under the* $\mathsf{MLWE}_{\mathcal{R},n,m,q,\bar{\chi}}$ *and* $\mathsf{MatrixHint\text{-}MLWE}^{N,\chi_1,\mathcal{S}}_{\mathcal{R},m+1,n,q,\chi_0}$ *assumptions. More precisely, for any PPT adversary $\mathcal{A}$, there exist PPT adversaries $\mathcal{B}_0$, $\mathcal{B}_1$ against MLWE assumption and Matrix Hint-MLWE assumption, such that*

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA}^{\mathsf{XR}}}_{\mathsf{PKE},N,\mathcal{A}}(\lambda) = \mathsf{Adv}^{\mathsf{MLWE}}_{\mathsf{para}_0,\mathcal{B}_0}(\lambda) + \mathsf{Adv}^{\mathsf{MatrixHint\text{-}MLWE}}_{\mathsf{para}_1,\mathcal{B}_1}(\lambda)$$

where $\mathsf{para}_0 := (\mathcal{R}, n, m, q, \bar{\chi})$ *and* $\mathsf{para}_1 := ((\mathcal{R}, m+1, n, q, \chi_0), (N, \chi_1, \mathcal{S}))$.

Proof. Let $\mathcal{A}$ be a PPT adversary against the $\mathsf{IND\text{-}CPA}^{\mathsf{XR}}$ security of our PKE as defined in Figure 3. We upper bound the advantage of $\mathcal{A}$ by the following games. Denote E_i as the event $\mathcal{A}$ wins Game_i. The games are described in Figure 22.

- Game_0: The game is the real $\mathsf{IND\text{-}CPA}^{\mathsf{XR}}$ security game shown in Figure 3 so that we have
$$\Pr[\mathsf{E}_0] = \Pr\left[\mathsf{GAME}^{\mathsf{IND\text{-}CPA}^{\mathsf{XR}}}_{\mathsf{PKE},N,\mathcal{A}}(\lambda) = 1\right].$$

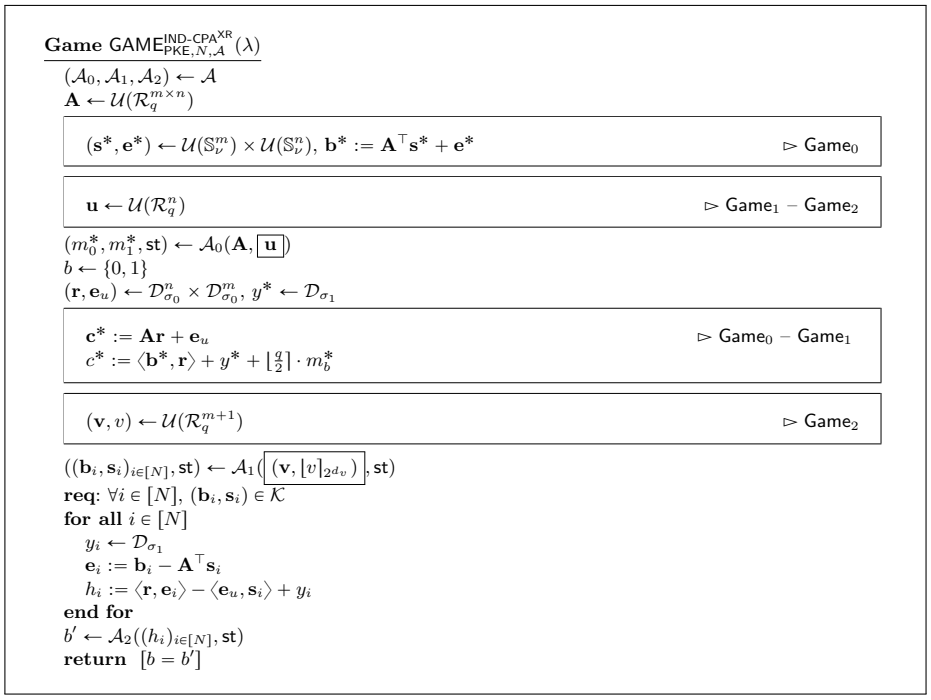

Fig. 22: The games for the proof of Theorem G.4.

– Game_1: The game is the same as Game_0 except that the challenger replaces the public key $\mathbf{b}^*$ by the uniformly random values $\mathbf{u}$.
The public key $\mathbf{b}^*$ is honestly generated, satisfying

$$\mathbf{b}^* = \mathbf{A}^\top \mathbf{s}^* + \mathbf{e}^*$$

where $\mathbf{s}^* \leftarrow \bar{\chi}^m$ and $\mathbf{e}^* \leftarrow \bar{\chi}^n$.
Therefore, the adversary $\mathcal{A}$ cannot distinguish between the challenger's public key $\mathbf{b}^*$ and the uniformly random values $\mathbf{u}$ under the MLWE assumption. There exists an adversary $\mathcal{B}_0$ with about the same running time as that of $\mathcal{A}$ such that

$$|\Pr[\mathsf{E}_1] - \Pr[\mathsf{E}_0]| = \mathsf{Adv}^{\mathsf{MLWE}}_{\mathsf{para}_0, \mathcal{B}_0}(\lambda)$$

where $\mathsf{para}_0 := (\mathcal{R}, n, m, q, \bar{\chi})$.
– Game_2: The game is the same as Game_1 except that the challenger modifies how the challenge ciphertext $(\mathbf{c}^*, c^*)$ is generated.
At a high level, the challenger replaces the challenge ciphertext $(\mathbf{c}^*, c^*)$ by the uniformly random values $(\mathbf{v}, v) \leftarrow \mathcal{U}(\mathcal{R}_q^{m+1})$ and the hints $(h_i)_{i \in [N]}$ can be interpreted as the hints for the secret of Matrix Hint-MLWE assumption. We show how to reduce this modification to the Matrix Hint-MLWE assumption as follows.

Denote the column vector $\hat{\mathbf{y}} = (y_i)_{i\in[N]}$ which is the concatenations of y_i in row-wise. Denote the column vector $\tilde{\mathbf{r}}$ and row vector $\boldsymbol{\gamma}_i$ for each $i \in [N]$ as

$$\tilde{\mathbf{r}} := \begin{pmatrix} y^* \\ \mathbf{e}_u \\ \mathbf{r} \end{pmatrix}, \quad \boldsymbol{\gamma}_i := \left(0 \,||\, -(\mathbf{s}_i)^\top \,||\, (\mathbf{e}_i)^\top\right)$$

and the hints can be rewritten as $h_i = \boldsymbol{\gamma}_i\tilde{\mathbf{r}} + y_i$ for $i \in [N]$. Denote the concatenation of $\boldsymbol{\gamma}_i$ and h_i for $i \in [N]$ in row-wise as $\mathbf{R} := (\boldsymbol{\gamma}_i)_{i\in[N]}$ and $\mathbf{h} := (h_i)_{i\in[N]}$ respectively, we have

$$\mathbf{h} = \mathbf{R}\tilde{\mathbf{r}} + \hat{\mathbf{y}}$$

where $\mathbf{R}$, $\tilde{\mathbf{r}}$, and $\hat{\mathbf{y}}$ are over the distributions of $\mathcal{S}$, χ_0, and χ_1 respectively. Note that the challenger will check the public-private key pairs provided by the adversary and if there exists $(\mathbf{s}_i^*, \mathbf{e}_i^*) \notin \mathbb{S}_\nu^m \times \mathbb{S}_\nu^n$, the challenger aborts the game and outputs $\perp$. Thus, $\mathbf{h}$ can be seen as the hint of secret vector $\tilde{\mathbf{r}}$ for the matrix $\mathbf{R}$ with $\ell := N$. And the challenge ciphertext $(\mathbf{c}^*, c^*)$ can be represented as

$$\left(\mathbf{I}_{m+1} \,\middle|\, \begin{matrix} \mathbf{u}^\top \\ \mathbf{A} \end{matrix}\right) \cdot \tilde{\mathbf{r}} + \begin{pmatrix} \lfloor \frac{q}{2} \rceil \cdot m_b^* \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} c^* \\ \mathbf{c}^* \end{pmatrix}.$$

It leads that even the adversary $\mathcal{A}$ can get the hint vector $\mathbf{h}$, the MLWE instance of $\tilde{\mathbf{r}}$, i.e., $(\mathbf{c}^*, c^*)$, is still indistinguished to the uniformly random values $(\mathbf{v}, v) \leftarrow \mathcal{U}(\mathcal{R}_q^{m+1})$ under $\mathsf{MatrixHint\text{-}MLWE}^{k,\chi_1,\mathcal{S}}_{\mathcal{R},m+1,n,q,\chi_0}$ assumption. Therefore, there exists an adversary $\mathcal{B}_1$ with about the same running time as that of $\mathcal{A}$ such that

$$|\Pr[\mathsf{E}_2] - \Pr[\mathsf{E}_1]| = \mathsf{Adv}^{\mathsf{MatrixHint\text{-}MLWE}}_{\mathsf{para}_1,\mathcal{B}_1}(\lambda)$$

where $\mathsf{para}_1 := ((\mathcal{R}, m+1, n, q, \chi_0), (N, \chi_1, \mathcal{S}))$.
Furthermore, in Game_2, the ciphertext output by the challenger is independent of the challenge bit b and therefore we have

$$\Pr[\mathsf{E}_2] = \frac{1}{2}.$$

Collecting all the games from Game_0 to Game_3, we get the required bound. □

G.4 Proofs for XR-KEM

Extended Reproducibility. We show the extended reproducibility of our construction as follows.

Theorem G.7 (Extended Reproducibility). *For any positive integer N, our* KEM *in Construction 4.8 is extended reproducible. More precisely, for the extended reproducible game in Figure 2, the following probability holds,*

$$\Pr\left[\mathsf{Game}^{\mathsf{ext\text{-}repr}}_{\mathsf{KEM},\mathsf{Rep},N}(\lambda) = 1\right] = 1.$$

Proof. The proof is analogous to Theorem 4.5. Suppose $(\mathsf{ct}^* := (\mathbf{c}, u^*), \mathsf{K}^* := \mu^*) \leftarrow \mathsf{Encap}(\mathbf{A}, \mathbf{b}^*)$ with randomness $\mathsf{r}_0 := (\mathbf{r}, \mathbf{e}_u)$, $\hat{\mathsf{r}}^* := y^*$, where we have

$$\mathbf{c} = \mathbf{A}\mathbf{r} + \mathbf{e}_u, \tag{G.18}$$

$u^* = \big\langle \mathsf{dbl}\big(\langle \mathbf{b}^*, \mathbf{r}\rangle + y^*\big)\big\rangle_2$, and $\mu^* = \big\lfloor \mathsf{dbl}\big(\langle \mathbf{b}^*, \mathbf{r}\rangle + y^*\big)\big\rceil_2$. For each $i \in [N]$, the public key $\mathbf{b}_i \leftarrow \mathsf{KGen}(\mathbf{A})$. Thus, we have

$$\mathbf{b}_i = \mathbf{A}^\top \mathbf{s}_i + \mathbf{e}_i. \tag{G.19}$$

For the hints $(h_i)_{i\in[N]} \leftarrow \mathsf{HintGen}((\mathbf{r}, \mathbf{e}_u), (\mathbf{b}_i, \mathbf{e}_i)_{i\in[N]}, (y_i)_{i\in[N]})$, we have

$$h_i = \langle \mathbf{r}, \mathbf{e}_i\rangle - \langle \mathbf{e}_u, \mathbf{s}_i\rangle + y_i. \tag{G.20}$$

On input h_i, $\mathsf{Rep}((\mathbf{c}, u^*), \mathbf{b}_i, \mathbf{s}_i, h_i)$ outputs the reproduced ciphertext $(\mathbf{c}, u_i)$ for $\mathbf{b}_i$, where

$$u_i = \big\langle \mathsf{dbl}\big(\langle \mathbf{c}, \mathbf{s}_i\rangle + h_i\big)\big\rangle_2. \tag{G.21}$$

When plugging Equation (G.18), Equation (G.19), Equation (G.20) into Equation (G.21), we have

$$u_i = \big\langle \mathsf{dbl}\big(\langle \mathbf{b}_i, \mathbf{r}\rangle + y_i\big)\big\rangle_2, \quad \mu_i = \big\lfloor \mathsf{dbl}\big(\langle \mathbf{b}_i, \mathbf{r}\rangle + y_i\big)\big\rceil_2$$

which are the same as the outputs from $\mathsf{Encap}(\mathbf{A}, \mathbf{b}_i; (\mathbf{r}, \mathbf{e}_u), y_i)$.

Overall, we get the extended reproducibility of our construction. □

Correctness. We set $\mathsf{Compress}(x) = \lfloor x \mod q \rceil_{2^{d_u}}$ and $\mathsf{Decompress}(x) = \lfloor x \mod 2^{d_u} \rceil_q$. Like our XR-PKE, here, we mainly consider the case that the key-independent ciphertext is compressed and then decompressed before the decryption, as done in mmKEM compiler of Construction B.2.

Theorem G.8 (Correctness). *Let $\mathbf{e}, \mathbf{s}, \mathbf{r}, \mathbf{e}_u, y$ be random variables that have the corresponding distribution as in Construction 4.8. Denote ζ as*

$$\Pr\left[\left\| 2\Big(\langle \mathbf{e}, \mathbf{r}\rangle + y - \langle \mathbf{s}, \mathbf{e}_u\rangle + \langle \mathbf{s}, \mathbf{c}_u\rangle\Big) - \bar{e}\right\|_\infty \geqslant \frac{q}{4}\right]$$

where $\mathbf{c}_u := \mathbf{c} - \lfloor \lfloor \mathbf{c} \bmod q \rceil_{2^{d_u}} \rceil_q \in \mathcal{R}^m$, and $\bar{e}$ denotes the error in $\mathsf{dbl}(c)$ function. We say our Construction 4.8 is ζ-correct.

Proof. Considering the compression and decompression of *independent* ciphertext $\mathbf{c}$, the value $\mathbf{c}$ (renamed as $\mathbf{c}'$) in Decap algorithm is

$$\mathbf{c}' := \lfloor \lfloor \mathbf{c} \mod q \rceil_{2^{d_u}} \rceil_q \ .$$

One can observe that the decapsulation is made via reconciliation mechanism. It means that the decapsulation succeeds if and only if the following equation holds,

$$\lfloor \bar{c} \rceil_2 = \mathsf{rec}\big(2 \cdot \langle \mathbf{c}', \mathbf{s}\rangle, \langle \bar{c} \rangle_2\big).$$

By Lemma 2.2, $\mathsf{rec}(\cdot,\cdot)$ works if the following holds,

$$\| \bar{c} - 2 \cdot \langle \mathbf{c}', \mathbf{s} \rangle \mod 2q \|_\infty < \frac{2q}{8} = \frac{q}{4}.$$

Plugging $\bar{c} = \mathsf{dbl}(c) = 2c - \bar{e}$ and $\mathbf{c}' = \mathbf{c} - \mathbf{c}_u$, the above inequality is equivalent to

$$\| 2c - \bar{e} - 2 \cdot \langle \mathbf{c} - \mathbf{c}_u, \mathbf{s} \rangle \|_\infty < \frac{q}{4}.$$

Since the value of $c := \langle \mathbf{b}, \mathbf{r} \rangle + y$ in $\mathsf{Encap}^{\mathsf{d}}$ algorithm where the value of $\mathbf{b} := \mathbf{A}^\top \mathbf{s} + \mathbf{e}$, we can obtain

$$2c - \bar{e} - 2 \cdot \langle \mathbf{c} - \mathbf{c}_u, \mathbf{s} \rangle = 2\Big(\langle \mathbf{e}, \mathbf{r} \rangle + y - \langle \mathbf{s}, \mathbf{e}_u \rangle + \langle \mathbf{s}, \mathbf{c}_u \rangle \Big) - \bar{e}.$$

It means that when ℓ_∞-norm of the decapsulation error is no less than $q/4$, i.e., $\|2(\langle \mathbf{e}, \mathbf{r} \rangle + y - \langle \mathbf{s}, \mathbf{e}_u \rangle + \langle \mathbf{c}_u, \mathbf{s} \rangle) - \bar{e}\|_\infty \geqslant q/4$, the decapsulation will fail. Thus, the value ζ is no more than the probability of decapsulation failure. □

Security. We show that our Construction 4.8 is $\mathsf{IND\text{-}CPA}^{\mathsf{XR}}$ secure if the MLWE assumption and the Matrix Hint-MLWE assumption are hard.

Theorem G.9 (Security). *Let m, n, d, q, N, ν be positive integers parameters. Let $\sigma, \sigma_0, \sigma_1$ be Gaussian width parameters. Let the positive real matrices Σ_1 and $\Sigma_{\mathbf{y}}$ be as Equation (4.5). Let the distribution $\mathcal{S}$ and the bound B be as Equation (4.3) and (4.4) respectively. Let the distribution $\chi_0 := \mathcal{D}_{\mathbb{Z}^{(m+n+N)d}, \sqrt{\Sigma_1}}$, $\chi_1 := \mathcal{D}_{\mathbb{Z}^{Nd}, \sqrt{\Sigma_{\mathbf{y}}}}$, $\bar{\chi} := \mathcal{U}(\mathbb{S}_\nu)$. Suppose Equation (4.1) and (4.2) hold.*

Our KEM *in Construction 4.8 is* $\mathsf{IND\text{-}CPA}^{\mathsf{XR}}$ *secure under the* $\mathsf{MLWE}_{\mathcal{R},n,m,q,\bar{\chi}}$ *and* $\mathsf{MatrixHint\text{-}MLWE}^{N,\chi_1,\mathcal{S}}_{\mathcal{R},m+1,n,q,\chi_0}$ *assumptions. More precisely, for any PPT adversary $\mathcal{A}$, there exist PPT adversaries $\mathcal{B}_0$, $\mathcal{B}_1$ against MLWE assumption and Matrix Hint-MLWE assumption, such that*

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA}^{\mathsf{XR}}}_{\mathsf{KEM},N,\mathcal{A}}(\lambda) = 2\left(\mathsf{Adv}^{\mathsf{MLWE}}_{\mathsf{para}_0,\mathcal{B}_0}(\lambda) + \mathsf{Adv}^{\mathsf{MatrixHint\text{-}MLWE}}_{\mathsf{para}_1,\mathcal{B}_1}(\lambda) \right)$$

where $\mathsf{para}_0 := (\mathcal{R}, n, m, q, \bar{\chi})$ *and* $\mathsf{para}_1 := ((\mathcal{R}, m+1, n, q, \chi_0), (N, \chi_1, \mathcal{S}))$.

Proof. The proof is analogous to Theorem 4.7. Let $\mathcal{A}$ be a PPT adversary against the $\mathsf{IND\text{-}CPA}^{\mathsf{XR}}$ security of our KEM as defined in Figure 15. We upper bound the advantage of $\mathcal{A}$ by the following games where the first and last game are $\mathsf{GAME}^{\mathsf{IND\text{-}CPA}^{\mathsf{XR}}}_{\mathsf{KEM},N,0,\mathcal{A}}(\lambda)$ and $\mathsf{GAME}^{\mathsf{IND\text{-}CPA}^{\mathsf{XR}}}_{\mathsf{KEM},N,1,\mathcal{A}}(\lambda)$, respectively. Denote E_i as the event that $\mathcal{A}$ wins the game Game_i. Game_0 – Game_3 are described in Figure 23.

- Game_0: The game is the real $\mathsf{IND\text{-}CPA}^{\mathsf{XR}}$ security game shown in Figure 15 with the challenger bit $b = 0$ so that we have

$$\Pr[\mathsf{E}_0] = \Pr\left[\mathsf{GAME}^{\mathsf{IND\text{-}CPA}^{\mathsf{XR}}}_{\mathsf{KEM},N,0,\mathcal{A}}(\lambda) = 1 \right].$$

- Game_1: The game is the same as Game_0 except that the challenger replaces the public key $\mathbf{b}^*$ by the uniformly random values $\mathbf{u} \leftarrow \mathcal{U}(\mathcal{R}_q^n)$.

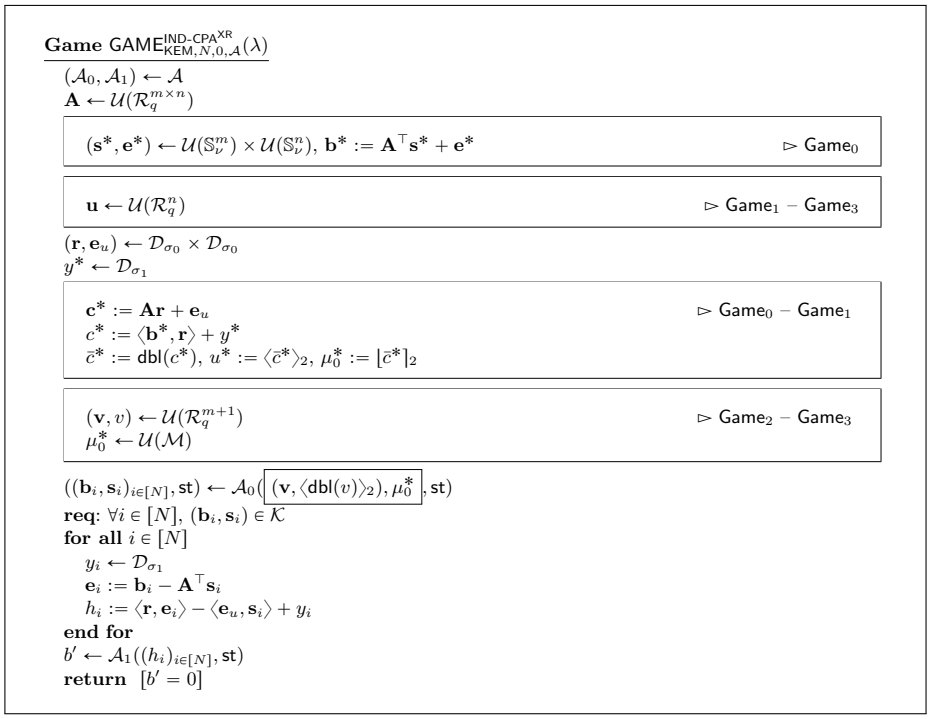

Fig. 23: Game_0 – Game_3 for the proof of Theorem G.9.

The public key $\mathbf{b}^*$ is honestly generated by the challenger, satisfying

$$\mathbf{b}^* = \mathbf{A}^\top \mathbf{s}^* + \mathbf{e}^* \tag{G.22}$$

where $\mathbf{s}^* \leftarrow \bar{\chi}^m$ and $\mathbf{e}^* \leftarrow \bar{\chi}^n$.
Therefore, the adversary $\mathcal{A}$ cannot distinguish between the challenger's public key $\mathbf{b}^*$ and the uniformly random values $\mathbf{u}$ under the MLWE assumption. There exists an adversary $\mathcal{B}_0$ with about the same running time as that of $\mathcal{A}$ such that

$$|\Pr[\mathsf{E}_1] - \Pr[\mathsf{E}_0]| = \mathsf{Adv}^{\mathsf{MLWE}}_{\mathsf{para}_0, \mathcal{B}_0}(\lambda)$$

where $\mathsf{para}_0 := (\mathcal{R}, n, m, q, \bar{\chi})$

– Game_2: The game is the same as Game_1 except that the challenger modifies how the challenge ciphertext $(\mathbf{c}^*, c^*)$ is generated.
At a high level, the challenger replaces the challenge ciphertext $(\mathbf{c}^*, c^*)$ by the uniformly random values $(\mathbf{v}, v) \leftarrow \mathcal{U}(\mathcal{R}_q^{m+1})$ and the hints $(h_i)_{i \in [N]}$ can be interpreted as the hints for the secret of Matrix Hint-MLWE assumption. We show how to reduce this modification to the Matrix Hint-MLWE assumption as follows.

Denote the column vector $\hat{\mathbf{y}} = (y_i)_{i\in[N]}$ which is the concatenations of y_i in row-wise. Denote the column vector $\tilde{\mathbf{r}}$ and row vector $\boldsymbol{\gamma}_i$ for each $i \in [N]$ as

$$\tilde{\mathbf{r}} := \begin{pmatrix} y^* \\ \mathbf{e}_u \\ \mathbf{r} \end{pmatrix}, \quad \boldsymbol{\gamma}_i := \left(0 \;||\; -(\mathbf{s}_i)^\top \;||\; (\mathbf{e}_i)^\top\right)$$

and the hints can be rewritten as $h_i = \boldsymbol{\gamma}_i\tilde{\mathbf{r}} + y_i$ for $i \in [N]$. Denote the concatenation of $\boldsymbol{\gamma}_i$ and h_i for $i \in [N]$ in row-wise as $\mathbf{R} := (\boldsymbol{\gamma}_i)_{i\in[N]}$ and $\mathbf{h} := (h_i)_{i\in[N]}$ respectively, we have

$$\mathbf{h} = \mathbf{R}\tilde{\mathbf{r}} + \hat{\mathbf{y}}$$

where $\mathbf{R}$, $\tilde{\mathbf{r}}$, and $\hat{\mathbf{y}}$ are over the distributions of $\mathcal{S}$, χ_0, and χ_1 respectively. Note that the challenger will check the public-private key pairs provided by the adversary and if there exists $(\mathbf{s}_i^*, \mathbf{e}_i^*) \notin \mathbb{S}_\nu^m \times \mathbb{S}_\nu^n$, the challenger aborts the game and outputs $\perp$. Thus, $\mathbf{h}$ can be seen as the hint of secret vector $\tilde{\mathbf{r}}$ for the matrix $\mathbf{R}$ with $\ell := N$. And the challenge ciphertext $(\mathbf{c}^*, c^*)$ can be represented as

$$\left(\mathbf{I}_{m+1} \middle| \begin{matrix} \mathbf{u}^\top \\ \mathbf{A} \end{matrix}\right) \cdot \tilde{\mathbf{r}} = \begin{pmatrix} c^* \\ \mathbf{c}^* \end{pmatrix}.$$

It leads that even the adversary $\mathcal{A}$ can get the hint vector $\mathbf{h}$, the MLWE instance of $\tilde{\mathbf{r}}$, i.e., $(\mathbf{c}^*, c^*)$, is still indistinguished to the uniformly random values $(\mathbf{v}, v) \leftarrow \mathcal{U}(\mathcal{R}_q^{m+1})$ under $\mathsf{MatrixHint\text{-}MLWE}^{k,\chi_1,\mathcal{S}}_{\mathcal{R},m+1,n,q,\chi_0}$ assumption. Therefore, there exists an adversary $\mathcal{B}_1$ with about the same running time as that of $\mathcal{A}$ such that

$$|\Pr[\mathsf{E}_2] - \Pr[\mathsf{E}_1]| = \mathsf{Adv}^{\mathsf{MatrixHint\text{-}MLWE}}_{\mathsf{para}_1,\mathcal{B}_1}(\lambda)$$

where $\mathsf{para}_1 := ((\mathcal{R}, m+1, n, q, \chi_0), (N, \chi_1, \mathcal{S}))$.

– Game_3: The game is the same as Game_2 except that the challenger replaces the encapsulated key μ_0^* by the key uniformly randomly sampled from the encapsulated key space $\mathcal{M}$.
One can observe that in Game_2, the key-*dependent* ciphertext c^* is replaced by the uniformly random value $v \leftarrow \mathcal{U}(\mathcal{R}_q)$, then the encapsulated key is generated by $\mu_0^* := \lfloor \mathsf{dbl}(v) \rceil_2$.
Therefore, by Lemma 2.1, $\mu_0^* := \lfloor \mathsf{dbl}(v) \rceil_2$ is uniformly random conditioned on $u^* = \langle \mathsf{dbl}(v) \rangle_2$. Therefore, when we replace the encapsulated key by $\mu_0^* \leftarrow \mathcal{U}(\mathcal{M})$ where $\mathcal{M} := \mathcal{R}_2$, one can observe that

$$\Pr[\mathsf{E}_3] = \Pr[\mathsf{E}_2].$$

Note that right now, the way of generating the encapsulated key μ_0^* is exactly the same as μ_1^*. In the rest of games, we are going to switch to the real $\mathsf{IND\text{-}CPA}^{\mathsf{XR}}$ security game with challenge bit $b = 1$ following the same ways in the games from Game_1 to Game_2.

– Game_4: The game is the same as Game_3 except that the challenger modifies how the challenge ciphertext $(\mathbf{c}^*, c^*)$ is generated.
Specifically, the challenger generates the challenge ciphertext $(\mathbf{c}^*, c^*)$ as $\mathbf{c}^* := \mathbf{A}\mathbf{r} + \mathbf{e}_u$ and $c^* := \langle \mathbf{b}^*, \mathbf{r} \rangle + y^*$. As the proof in Game_2, we have

$$|\Pr[\mathsf{E}_4] - \Pr[\mathsf{E}_3]| = \mathsf{Adv}^{\mathsf{MatrixHint\text{-}MLWE}}_{\mathsf{para}_1, \mathcal{B}_1}(\lambda)$$

where $\mathsf{para}_1 := ((\mathcal{R}, m+1, n, q, \chi_0), (N, \chi_1, \mathcal{S}))$.
– Game_5: The game is the same as Game_4 except that the challenger changes how the public key $\mathbf{b}^*$ is generated.
Specifically, the challenger generates the public key $\mathbf{b}^*$ as Equation (G.22). As the proof in Game_1, we have

$$|\Pr[\mathsf{E}_5] - \Pr[\mathsf{E}_4]| = \mathsf{Adv}^{\mathsf{MLWE}}_{\mathsf{para}_0, \mathcal{B}_0}(\lambda)$$

where $\mathsf{para}_0 := (\mathcal{R}, n, m, q, \bar{\chi})$.
For now, one can observe that Game_5 is exactly the same as the real $\mathsf{IND\text{-}CPA}^{\mathsf{XR}}$ security game with challenge bit $b = 1$. Thus, we have

$$\Pr[\mathsf{E}_5] = \Pr\left[\mathsf{GAME}^{\mathsf{IND\text{-}CPA}^{\mathsf{XR}}}_{\mathsf{KEM}, N, 1, \mathcal{A}}(\lambda) = 1\right].$$

Collecting all the games from Game_0 to Game_5, we get the required bound. □

G.5 Security Proof for KOSK Compiler

We restate Theorem4.11 below and provide its formal proof.

Theorem G.10 (Security). *For* $\mathsf{ATK} \in \{\mathsf{CPA}, \mathsf{CCA}\}$*, if* mmPKE' *is* $\mathsf{mmIND\text{-}ATK}^{\mathsf{KOSK}}$ *secure and* Π *is a NIZK argument system satisfies correctness, multi-proof extractability and zero knowledge, our* $\mathsf{mmPKE} \leftarrow \mathsf{Comp}^{\mathsf{KOSK}}[\mathsf{mmPKE}', \Pi]$ *output by Construction 4.10 is* $\mathsf{mmIND\text{-}ATK}$ *secure.*

Proof. The proof is similar to [13, Theorem 8.3], especially on the use of multi-proof extractability. Suppose there is a PPT adversary $\mathcal{A} := (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ which wins the $\mathsf{mmIND\text{-}ATK}$ security game of mmPKE with non-negligible probability ϵ. Suppose $\mathcal{A}$ makes at most Q_{H} queries to the random oracles H. Without loss of generality, assume that $\mathcal{A}$ never repeats a random oracle query.

We prove the statement by introducing a sequence of games. Denote E_i as the event $\mathcal{A}$ wins Game_i. The games are described in Figure 24.

– Game_0: The game is the real $\mathsf{mmIND\text{-}ATK}$ security game of $\mathsf{mmPKE} \leftarrow \mathsf{Comp}^{\mathsf{KOSK}}[\mathsf{mmPKE}', \Pi]$ shown in Figure 11. Here, by definition we have

$$\Pr[\mathsf{E}_0] = \epsilon.$$

– Game_1: The game is the same as Game_0 except that we generate the proof $(\pi_i)_{i \in [\ell]}$ by the simulator Sim_1. It is easy to see that Game_1 and Game_0 are indistinguishable by the zero-knowledge property of Π, i.e., one can construct a PPT adversary $\mathcal{B}_0$ such that

$$\Pr[\mathsf{E}_1] \geqslant \Pr[\mathsf{E}_0] - \ell \cdot \mathsf{Adv}^{\mathsf{ZK}}_{\Pi, \mathcal{B}_0}(\lambda) = \Pr[\mathsf{E}_0] - \mathsf{negl}(\lambda).$$

– Game_2: The game is the same as Game_1 except that we program the output of $\mathsf{H}(0)$ from crs_Π to $\widetilde{\mathsf{crs}}_\Pi$ where $(\widetilde{\mathsf{crs}}_\Pi, \tau) \leftarrow \mathsf{Sim}_{\mathsf{crs}}(1^\lambda)$. It can be checked that Game_2 and Game_1 are indistinguishable by the CRS indistinguishability in multi-proof extractability. Specifically, there exists a PPT adversary $\mathcal{B}_1$ such that

$$\Pr[\mathsf{E}_2] \geqslant \Pr[\mathsf{E}_1] - \mathsf{Adv}^{\mathsf{crs}}_{\Pi,\mathcal{B}_1}(\lambda) = \Pr[\mathsf{E}_1] - \mathsf{negl}(\lambda).$$

– Game_3: The game is the same as Game_2 except that we use the multi-proof extractability of Π to extract the witnesses for all proofs $(\pi_i)_{i\in[\ell:N]}$ that are generated by the adversary $\mathcal{A}$. More precisely, the reduction will run

$$\mathsf{sk}_i \leftarrow \mathsf{Multi\text{-}Extract}(1^\lambda, Q_\mathsf{H}, Q_s, 1/\Pr[\mathsf{E}_2], \tau, \mathsf{pk}_i, \pi_i)$$

where $Q_\mathsf{H} = \mathsf{poly}(\lambda)$ is the number of the random oracle queries by the adversary $\mathcal{A}$ and $Q_s \leqslant N$ is the number of statement-proof pairs (pk_i, π_i) generated by the adversary $\mathcal{A}$.
Let $\mathsf{Abort}_{\mathsf{extract}}$ be the event that $(\mathsf{pk}_i, \mathsf{sk}_i) \notin R_\Pi$ for some $i \in [Q_s]$. If $\mathsf{Abort}_{\mathsf{extract}}$ occurs then the reduction aborts and overwrites the adversary's output to be $\perp$. We note that the reduction does not use the extracted witness in this game.
Arguing identically as in [59, Lemma 3.6] and assuming that $\Pr[\mathsf{E}_2]$ is non-negligible, the runtime of the reduction is still $\mathsf{poly}(\lambda)$ and also

$$\Pr[\mathsf{E}_3] \geqslant \frac{1}{2}\Pr[\mathsf{E}_2] - \mathsf{negl}(\lambda).$$

– Game_4: The game is the same as Game_3 except that we generate pp', $(\mathsf{pk}_i)_{i\in[\ell]}$, and $\mathbf{ct}$ by the challenger $\mathcal{C}$ in $\mathsf{mmIND\text{-}ATK}^{\mathsf{KOSK}}$ security game of mmPKE'. Specifically, we first forward the value ℓ from the adversary $\mathcal{A}$ to the challenger $\mathcal{C}$ and get $(\mathsf{pk}_i)_{i\in[\ell]}$ from the challenger $\mathcal{C}$. Then, we run the simulator to obtain $(\pi_i)_{i\in[\ell]}$. After sending them to $\mathcal{A}$, we can obtain $(\mathsf{pk}_i, \pi_i)_{i\in[\ell:N]}$ and $(\mathsf{m}_i^0, \mathsf{m}_i^1)_{i\in[\ell]}$, $(\mathsf{m}_i)_{i\in[\ell:N]}$ from $\mathcal{A}$. With the multi-proof extractor, the private key sk_i of the public key pk_i generated by $\mathcal{A}$ can be extracted. We send the extracted private key along with the public key, and the two challenge message vectors $(\mathsf{m}_i^0, \mathsf{m}_i^1)_{i\in[\ell]}$ and $(\mathsf{m}_i)_{i\in[\ell:N]}$ provided by $\mathcal{A}$ to $\mathcal{C}$ and receive the challenge ciphertext $\mathbf{ct}$ from $\mathcal{C}$. After forward $\mathbf{ct}$ to $\mathcal{A}$, we can obtain the guess bit b' from $\mathcal{A}$ and set the guess bit for $\mathcal{C}$.
One can observe Game_4 is the same as Game_3, i.e.,

$$\Pr[\mathsf{E}_4] = \Pr[\mathsf{E}_3]$$

and also Game_4 is the $\mathsf{mmIND\text{-}ATK}^{\mathsf{KOSK}}$ security game of mmPKE'. Thus, there exists an adversary $\mathcal{B}_2$ with about the same running time as that of $\mathcal{A}$ such that

$$\Pr[\mathsf{E}_4] = \mathsf{Adv}^{\mathsf{mmIND\text{-}ATK}^{\mathsf{KOSK}}}_{\mathsf{mmPKE}',N,\mathcal{B}_2}(\lambda)$$

Collecting all the games from Game_0 to Game_4, we get the $\mathsf{mmIND\text{-}ATK}$ security of mmPKE. □

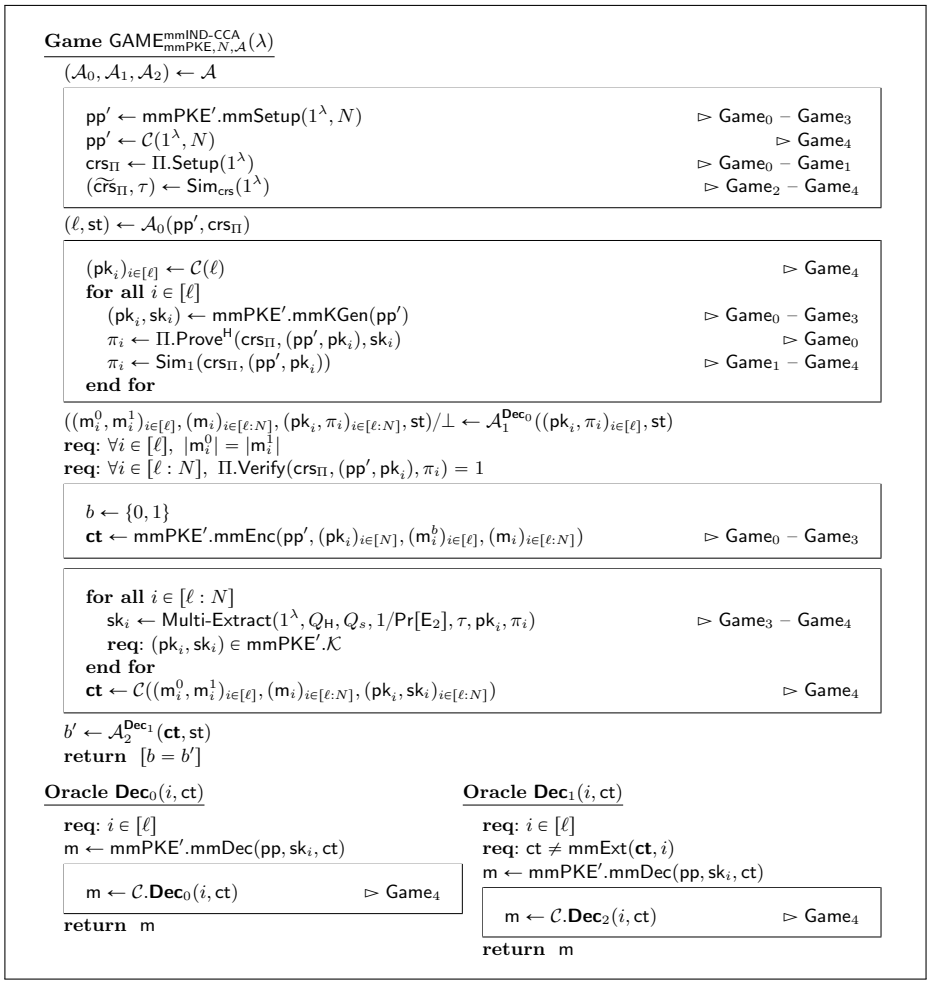

Fig. 24: Game_0 - Game_4 for the proof of Theorem G.10. For $\mathsf{ATK} = \mathsf{CPA}$, the adversary $\mathcal{A}$ does not have the access to decryption oracles $\mathbf{Dec}_0$ and $\mathbf{Dec}_1$.

G.6 Security Proof for Adaptive Security Compiler

We restate Theorem D.2 below and provide their formal proofs.

Theorem G.11 (Security). *If* mmPKE' *is* $\mathsf{mmIND\text{-}CPA}$ *secure and* Π' *is a NIZK argument system satisfies correctness, zero knowledge, and simulation soundness, our* $\mathsf{mmPKE} \leftarrow \mathsf{Comp}^{\mathsf{CCA}}[\mathsf{mmPKE}', \Pi']$ *output by Construction D.1 is* $\mathsf{mmIND\text{-}CCA}^{\mathsf{Cor}}$ *secure.*

Proof. Let $\mathcal{A}$ be an PPT adversary against the $\mathsf{mmIND\text{-}CCA}^{\mathsf{Cor}}$ security of mmPKE. We define the following sequence of games where the first and last

game are the game $\mathsf{GAME}^{\mathsf{mmIND\text{-}CCA}^{\mathsf{Cor}}}_{\mathsf{mmPKE},N,0,\mathcal{A}}(\lambda)$ and $\mathsf{GAME}^{\mathsf{mmIND\text{-}CCA}^{\mathsf{Cor}}}_{\mathsf{mmPKE},N,1,\mathcal{A}}(\lambda)$, respectively. Denote E_i as the event that $\mathcal{A}$ wins the game Game_i.

– Game_0: The game is the real security game $\mathsf{GAME}^{\mathsf{mmIND\text{-}CCA}^{\mathsf{Cor}}}_{\mathsf{mmPKE},N,0,\mathcal{A}}(\lambda)$ shown in Figure 11 with the challenge bit $b = 0$. It means that the challenger encrypts the messages $(\mathsf{m}_i^0)_{i\in[\ell]}$ and $(\mathsf{m}_i)_{i\in[\ell:N]}$ to the challenge ciphertext $\mathbf{ct}$.

$$\Pr[\mathsf{E}_0] = \Pr\left[\mathsf{GAME}^{\mathsf{mmIND\text{-}CCA}^{\mathsf{Cor}}}_{\mathsf{mmPKE},N,0,\mathcal{A}}(\lambda) = 1\right].$$

– Game_1: The game is the same as Game_0 except that the challenger simulates the proof $(\pi_i)_{i\in[N]}$ in the ciphertext $\mathbf{ct}$ by the simulator Sim_1 as shown in Figure 25.
Hence, there exists a reduction $\mathcal{B}_0$ to the computational zero knowledge of Π' such that

$$|\Pr[\mathsf{E}_1] - \Pr[\mathsf{E}_0]| \leqslant N \cdot \mathsf{Adv}^{\mathsf{ZK}}_{\Pi',\mathcal{B}_0}(\lambda).$$

– Game_2: The game is the same as Game_1 except that the challenger switches $(\mathsf{m}_i^0)_{i\in[N]}$ to $(\mathsf{m}_i^1)_{i\in[N]}$ in $(\widehat{\mathsf{ct}}_0^{(i)})_{i\in[N]}$, the first key-*dependent* ciphertext of $(\widehat{\mathsf{ct}}_i := (\widehat{\mathsf{ct}}_0^{(i)}, \widehat{\mathsf{ct}}_1^{(i)}))_{i\in[N]}$, as shown in Figure 25. Note that here we set $\mathsf{m}_i^0 = \mathsf{m}_i^1 = \mathsf{m}_i$ for $i \in [\ell : N]$ to simplify the presentation.

$\mathsf{r}_0 \leftarrow \mathcal{D}_\mathsf{i}$, $\mathsf{ct}_0 \leftarrow \mathsf{mmPKE}'.\mathsf{mmEnc}^\mathsf{i}(\mathsf{pp}; \mathsf{r}_0)$
$\vec{\beta} := (\beta_i)_{i\in[N]} \leftarrow \{0,1\}^N$
for all $i \in [N]$
  $\hat{\mathsf{r}}_0^{(i)}, \hat{\mathsf{r}}_1^{(i)} \leftarrow D_\mathsf{d}$
  $\widehat{\mathsf{ct}}_0^{(i)} \leftarrow \mathsf{mmPKE}'.\mathsf{mmEnc}^\mathsf{d}(\mathsf{pp}, \mathsf{pk}_{\beta_i}^{(i)}, \boxed{\mathsf{m}_i^1}; \mathsf{r}_0, \hat{\mathsf{r}}_{\beta_i}^{(i)})$ ▷ Game_2
  $\widehat{\mathsf{ct}}_1^{(i)} \leftarrow \mathsf{mmPKE}'.\mathsf{mmEnc}^\mathsf{d}(\mathsf{pp}, \mathsf{pk}_{1-\beta_i}^{(i)}, \boxed{\mathsf{m}_i^0}; \mathsf{r}_0, \hat{\mathsf{r}}_{1-\beta_i}^{(i)})$ ▷ Game_2
  $\boxed{\pi_i \leftarrow \mathrm{Sim}_1(\mathsf{crs}_{\Pi'}, (\mathsf{pp}', \mathsf{pk}_0^{(i)}, \mathsf{pk}_1^{(i)}, \widehat{\mathsf{ct}}, \widehat{\mathsf{ct}}_0^{(i)}, \widehat{\mathsf{ct}}_1^{(i)}, \beta_i))}$ ▷ Game_1 - Game_2
  $\widehat{\mathsf{ct}}_i := (\widehat{\mathsf{ct}}_0^{(i)}, \widehat{\mathsf{ct}}_1^{(i)})$
end for
return $\mathbf{ct} := (\mathsf{ct}_0, (\widehat{\mathsf{ct}}_i)_{i\in[N]}, \vec{\beta}, (\pi_i)_{i\in[N]})$

Fig. 25: Game_1 and Game_2 for the proof of Theorem G.11.

Let BAD be the event that the adversary $\mathcal{A}$ can make a valid but improper query (e.g., double encryption for different message) to the decryption oracle (different from the challenge ciphertext $\mathbf{ct}$). If BAD happens, we abort the reduction. We claim that there exists an reduction algorithm $\mathcal{B}_1$ whose running time is about the same as $\mathcal{A}$, such that

$$|\Pr[\mathsf{E}_2] - \Pr[\mathsf{E}_1]| \leqslant \mathsf{Adv}^{\mathsf{mmIND\text{-}CPA}}_{\mathsf{mmPKE}',2N,\mathcal{B}_1}(\lambda) + \Pr[\mathsf{BAD}].$$

The reduction $\mathcal{B}_1$ is described in Figure 26.

The proof is a combination between the proof in [5, 33, 35] and [51, 61]. Roughly, $\mathcal{B}_1$ combines two key-*dependent* ciphertext of mmPKE' to form the ciphertext of mmPKE, which one is encrypted by the public keys from $\mathcal{B}_1$'s challenger and the other is encrypted by the public keys from $\mathcal{B}_1$ itself. $\mathcal{B}_1$ will switch the message $(\mathsf{m}_i^0)_{i\in[N]}$ to $(\mathsf{m}_i^1)_{i\in[N]}$ in the key-*dependent* ciphertext encrypted by its challenger's public key. If $\mathcal{A}$ can identify the modification, $\mathcal{B}_1$ can utilize $\mathcal{A}$ to break the $\mathsf{mmIND\text{-}CPA}$ security of mmPKE'.

Specifically, after receiving ℓ public keys $(\mathsf{pk}_i^*)_{i\in[\ell]}$ from the challenger of mmPKE', $\mathcal{B}_1$ picks these ℓ public keys as the part of the public keys for mmPKE and generates the rest ℓ public-private key pair $(\mathsf{pk}_i', \mathsf{sk}_i')_{i\in[\ell]}$ by itself. To decide which one of the two pubic keys in each public keys pk_i of mmPKE is from the challenger, $\mathcal{B}_1$ tosses a random bit α_i: if $\alpha_i = 0$, then $\mathsf{pk}_i := (\mathsf{pk}_i^*, \mathsf{pk}_i')$; otherwise, $\mathsf{pk}_i := (\mathsf{pk}_i', \mathsf{pk}_i^*)$. Then, like Game_1, $\mathcal{B}_1$ runs the simulator to get $(\mathsf{pp}_{\Pi'}, \tau) \leftarrow \mathsf{Sim}_0$ and sends the public parameter along with the public keys to the adversary $\mathcal{A}$.

To handle the corruption query, $\mathcal{B}_1$ can just flip the random bit α_i in the private key and provide the private key corresponding to public key generated by itself as the respond. And the adversary $\mathcal{A}$ cannot distinguish between the two public key since the random bit α_i in each uncorrupted private key is information-theoretically hiding to $\mathcal{A}$.

To handle the decryption query, $\mathcal{B}_1$ can use the private key generated by itself to decrypt the ciphertext. If BAD does not happen, it means that the adversary $\mathcal{A}$ cannot generate a valid proof for a ciphertext with different message to distinguish between the two public keys, even after seeing the simulated proof in the challenge ciphertext $\mathbf{ct}$. Thus, we can bound $\Pr[\mathsf{BAD}]$ by constructing a reduction $\mathcal{B}_2$ to the computational simulation soundness of Π', i.e.,

$$\Pr[\mathsf{BAD}] \leqslant Q_D \cdot \mathsf{Adv}^{\mathsf{SS}}_{\Pi',\mathcal{B}_2}(\lambda)$$

where Q_D denotes the number of the adversary $\mathcal{A}$'s queries to the decryption oracles $\mathbf{Dec}_0$ and $\mathbf{Dec}_1$.

To encrypt the challenge ciphertext, after receiving the public keys and message chosen by the adversary $\mathcal{A}$, $\mathcal{B}_1$ set $\vec{\beta} := \vec{\alpha}$ for switching the public keys during the encryption. It leads that the first key-*dependent* ciphertext $\widehat{\mathsf{ct}}_0^{(i)}$ in each key-*dependent* ciphertext $(\widehat{\mathsf{ct}}_0^{(i)}, \widehat{\mathsf{ct}}_1^{(i)})$ of mmPKE is encrypted by the challenger's public key. Since these cases are exclusive, α_i or β_i is uniformly random in $\mathcal{A}$'s view. After sending the public keys along with the two message vectors to its challenger, $\mathcal{B}_1$ obtains the ciphertext from its challenger. Like Game_1, $\mathcal{B}_1$ runs the simulator to obtain the proof $\pi_i \leftarrow \mathsf{Sim}_1$ for each $i \in [N]$. The challenge ciphertext with the proofs are sent to the adversary $\mathcal{A}$.

In the end, $\mathcal{B}_1$ uses the guess bit b' from $\mathcal{A}$ to break the $\mathsf{mmIND\text{-}CPA}$ security of mmPKE'. Thus, if mmPKE' is $\mathsf{mmIND\text{-}CPA}$ secure, the adversary $\mathcal{A}$ cannot know whether $\mathcal{B}_1$ switches the message m_0 to m_1 or not in the first key-*dependent* ciphertext. We get the above bound.

– Game_3: The game is the same as Game_2 except that the challenger switches $(\mathsf{m}_i^0)_{i\in[N]}$ to $(\mathsf{m}_i^1)_{i\in[N]}$ in $(\widehat{\mathsf{ct}}_1^{(i)})_{i\in[N]}$, the second key-*dependent* ciphertext of $(\widehat{\mathsf{ct}}_i := (\widehat{\mathsf{ct}}_0^{(i)}, \widehat{\mathsf{ct}}_1^{(i)}))_{i\in[N]}$.
If mmPKE' is $\mathsf{mmIND\text{-}CPA}$ secure, the adversary $\mathcal{A}$ is indistinguished between Game_2 and Game_3. We claim that there exists an reduction algorithm $\mathcal{B}_3$ whose running time is about the same as $\mathcal{A}$, such that

$$|\Pr[\mathsf{E}_3] - \Pr[\mathsf{E}_2]| \leqslant \mathsf{Adv}^{\mathsf{mmIND\text{-}CPA}}_{\mathsf{mmPKE}',2N,\mathcal{B}_3}(\lambda) + \Pr[\mathsf{BAD}].$$

The reduction $\mathcal{B}_3$ is analogous to $\mathcal{B}_1$ in Game_1 except that the challenger's public key is the second one in the public key of mmPKE.
– Game_4: The game is the same as Game_3 except that the challenger generates the proof $(\pi_i)_{i\in[N]}$ in the ciphertext $\mathbf{ct}$ by $\Pi'.\mathsf{Prove}$. Hence, there exists a reduction $\mathcal{B}_4$ to the computational zero knowledge of Π' such that

$$|\Pr[\mathsf{E}_4] - \Pr[\mathsf{E}_3]| \leqslant N \cdot \mathsf{Adv}^{\mathsf{ZK}}_{\Pi',\mathcal{B}_4}(\lambda).$$

Finally, Game_4 is the $\mathsf{mmIND\text{-}CCA}^{\mathsf{Cor}}$ security game with the challenge bit $b = 1$. And if the honestly generated proof π_i is not valid, the reduction aborts. Thus, we have

$$\Pr[\mathsf{E}_4] = \Pr\left[\mathsf{GAME}^{\mathsf{mmIND\text{-}CCA}^{\mathsf{Cor}}}_{\mathsf{mmPKE},N,1,\mathcal{A}}(\lambda) = 1\right].$$

Collecting all the games from Game_0 to Game_4, we get the $\mathsf{mmIND\text{-}CCA}^{\mathsf{Cor}}$ security of mmPKE. □

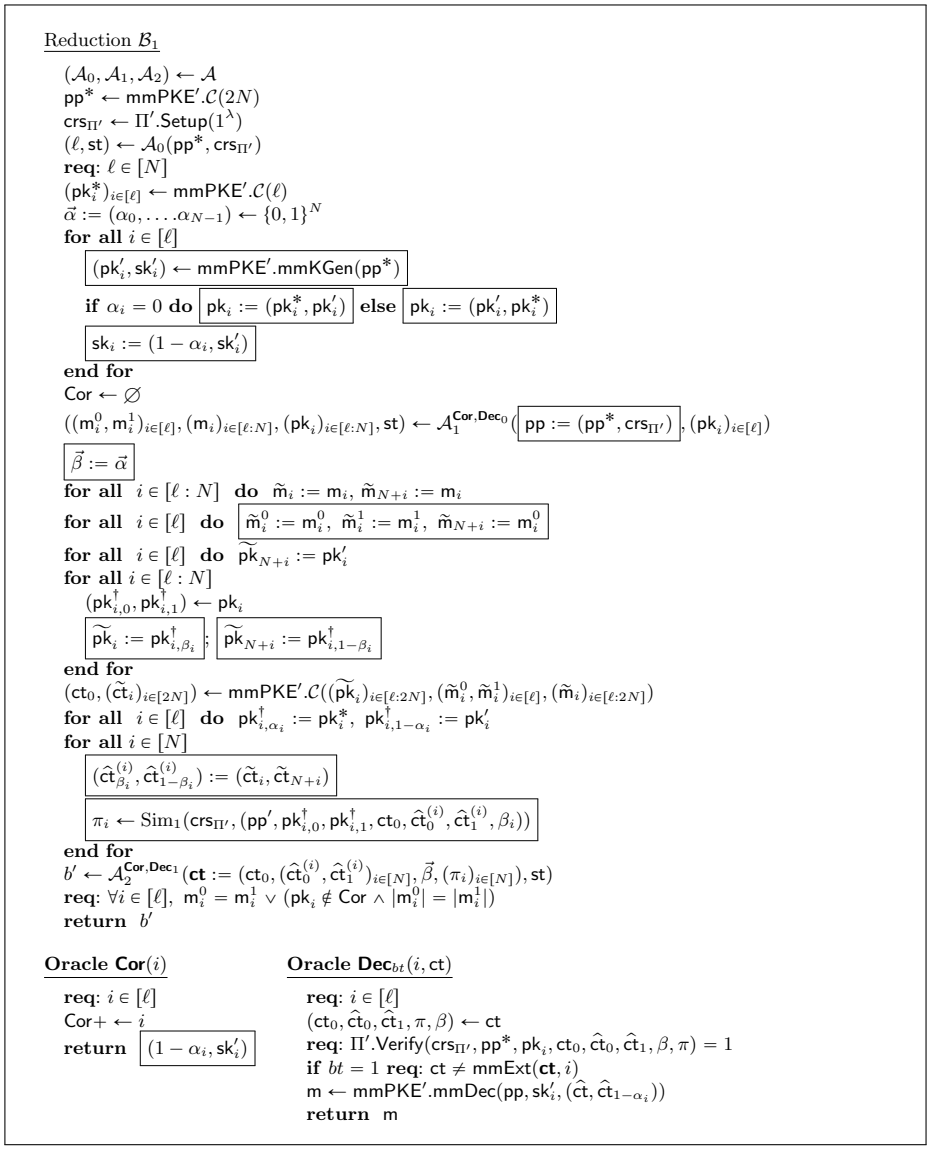

Fig. 26: The reduction $\mathcal{B}_1$ using a distinguisher $\mathcal{A}$ between Game_1 and Game_2 to break the mmIND-CPA security of mmPKE' in Theorem G.11. $\mathbf{Dec}_{bt}$ oracle is assigned to $\mathcal{A}_{bt}$ for $bt \in \{0,1\}$. The parts where $\mathcal{B}_1$'s operations are different from $\mathsf{mmIND\text{-}CCA}^{\mathsf{Cor}}$ security game are marked by boxes.