Tightly Secure CCA-Secure Encryption without Pairings

Romain Gay[1,⋆], Dennis Hofheinz[2,⋆⋆], Eike Kiltz[3,⋆⋆⋆], and Hoeteck Wee[1,†]

[1] ENS, Paris, France
rgay,wee@di.ens.fr

[2] Ruhr-Universität Bochum, Bochum, Germany
eike.kiltz@rub.de

[3] Karlsruhe Institute of Technology, Karlsruhe, Germany
Dennis.Hofheinz@kit.edu

Abstract. We present the first CCA-secure public-key encryption scheme based on DDH where the security loss is independent of the number of challenge ciphertexts and the number of decryption queries. Our construction extends also to the standard k-Lin assumption in pairing-free groups, whereas all prior constructions starting with Hofheinz and Jager (Crypto '12) rely on the use of pairings. Moreover, our construction improves upon the concrete efficiency of existing schemes, reducing the ciphertext overhead by about half (to only 3 group elements under DDH), in addition to eliminating the use of pairings.
We also show how to use our techniques in the NIZK setting. Specifically, we construct the first tightly simulation-sound designated-verifier NIZK for linear languages without pairings. Using pairings, we can turn our construction into a highly optimized publicly verifiable NIZK with tight simulation-soundness.

1 Introduction

The most basic security guarantee we require of a public key encryption scheme is that of semantic security against chosen-plaintext attacks (CPA) [15]: it is infeasible to learn anything about the plaintext from the ciphertext. On the other hand, there is a general consensus within the cryptographic research community that in virtually every practical application, we require semantic security against adaptive chosen-ciphertext attacks (CCA) [32, 13], wherein an adversary is given access to decryptions of ciphertexts of her choice.

In this work, we focus on the issue of security reduction and security loss in the construction of CPA and CCA-secure public-key encryption from the DDH assumption. Suppose we have such a scheme along with a security reduction showing that attacking the scheme in time t with success probability ϵ implies breaking the DDH assumption in time roughly t with success probability ϵ/L; we refer to L as the security loss. In general, L would depend on the security parameter λ as well as the number of challenge ciphertexts Q_{enc} and the number decryption queries Q_{dec}, and we say that we have a *tight security reduction* if L depends only on the security parameter and is independent of both Q_{enc} and Q_{dec}. Note that for typical settings of parameters (e.g., $\lambda = 80$ and $Q_{\mathsf{enc}}, Q_{\mathsf{dec}} \approx 2^{20}$, or even $Q_{\mathsf{enc}}, Q_{\mathsf{dec}} \approx 2^{30}$ in truly large settings), λ is much smaller than Q_{enc} and Q_{dec}.

In the simpler setting of CPA-secure encryption, the ElGamal encryption scheme already has a tight security reduction to the DDH assumption [29, 6], thanks to random self-reducibility of DDH

⋆ CNRS, INRIA. Supported by ERC Project aSCEND (639554).
⋆⋆ Supported by DFG grants HO 4534/2-2, HO 4534/4-1.
⋆⋆⋆ Partially supported by DFG grant KI 795/4-1 and ERC Project ERCC (FP7/615074).
† CNRS (UMR 8548), INRIA and Columbia University. Partially supported by the Alexander von Humboldt Foundation, NSF Award CNS-1445424 and ERC Project aSCEND (639554).

with a tight security reduction. In the case of CCA-secure encryption, the best result is still the seminal Cramer-Shoup encryption scheme [11], which achieves security loss Q_{enc}.[4] This raises the following open problem:

> Does there exist a CCA-secure encryption scheme with a tight security reduction to the DDH assumption?

Hofheinz and Jager [17] gave an affirmative answer to this problem under stronger (and pairing-related) assumptions, notably the 2-Lin assumptions in bilinear groups, albeit with large ciphertexts and secret keys; a series of follow-up works [24, 26, 5, 16] leveraged techniques introduced in the context of tightly-secure IBE [10, 7, 19] to reduce the size of ciphertext and secret keys to a relatively small constant. However, all of these works rely crucially on the use of pairings, and seem to shed little insight on constructions under the standard DDH assumption; in fact, a pessimist may interpret the recent works as strong indication that the use of pairings is likely to be necessary for tightly CCA-secure encryption.

We may then restate the open problem as eliminating the use of pairings in these prior CCA-secure encryption schemes while still preserving a tight security reduction. From a theoretical standpoint, this is important because an affirmative answer would yield tightly CCA-secure encryption under qualitatively weaker assumptions, and in addition, shed insight into the broader question of whether tight security comes at the cost of qualitative stronger assumptions.

Eliminating the use of pairings is also important in practice as it allows us to instantiate the underlying assumption over a much larger class of groups that admit more efficient group operations and more compact representations, and also avoid the use of expensive pairing operations. Similarly, tight reductions matter in practice because as L increases, we should increase the size of the underlying groups in order to compensate for the security loss, which in turn increases the running time of the implementation. Note that the impact on performance is quite substantial, as exponentiation in a r-bit group takes time roughly $\mathcal{O}(r^3)$.

1.1 Our Results

We settle the main open problem affirmatively: we construct a tightly CCA-secure encryption scheme from the DDH assumption without pairings. Moreover, our construction improves upon the concrete efficiency of existing schemes, reducing the ciphertext overhead by about half, in addition to eliminating the use of pairings. We refer to Figure 2 for a comparison with prior works.

Overview of our construction. Fix an additively written group $\mathbb{G}$ of order q. We rely on implicit representation notation [14] for group elements: for a fixed generator P of $\mathbb{G}$ and for a matrix $\mathbf{M} \in \mathbb{Z}_q^{n\times t}$, we define $[\mathbf{M}] := \mathbf{M}P \in \mathbb{G}^{n\times t}$ where multiplication is done component-wise. We rely on the $\mathcal{D}_k$-MDDH Assumption [14], which stipulates that given $[\mathbf{M}]$ drawn from a matrix distribution $\mathcal{D}_k$ over $\mathbb{Z}_q^{(k+1)\times k}$, $[\mathbf{Mx}]$ is computationally indistinguishable from a uniform vector in $\mathbb{G}^k$; this is a generalization of the k-Lin Assumption.

We outline the construction under the k-Lin assumption over $\mathbb{G}$, of which the DDH assumption is a special case corresponding to $k = 1$.

In this overview, we will consider a weaker notion of security, namely tag-based KEM security against plaintext check attacks (PCA) [31]. In the PCA security experiment, the adversary gets

[4] We ignore contributions to the security loss that depend only on a statistical security parameter.

no decryption oracle (as with CCA security), but a PCA oracle that takes as input a tag and a ciphertext/plaintext pair and checks whether the ciphertext decrypts to the plaintext. Furthermore, we restrict the adversary to only query the PCA oracle on tags different from those used in the challenge ciphertexts. PCA security is strictly weaker than the CCA security we actually strive for, but allows us to present our solution in a clean and simple way. (We show how to obtain full CCA security separately.)

The starting point of our construction is the Cramer-Shoup KEM, in which $\mathsf{Enc}_{\mathsf{KEM}}(\mathsf{pk}, \tau)$ outputs the ciphertext/plaintext pair

$$([\mathbf{y}], [z]) = ([\mathbf{x}^\top \mathbf{M}^\top], [\mathbf{x}^\top \mathbf{M}^\top \mathbf{k}_\tau]), \tag{1}$$

where $\mathbf{k}_\tau = \mathbf{k}_0 + \tau \mathbf{k}_1$ and $\mathsf{pk} := ([\mathbf{M}], [\mathbf{M}^\top \mathbf{k}_0], [\mathbf{M}^\top \mathbf{k}_1])$ for $\mathbf{M} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{(k+1)\times k}$. The KEM is PCA-secure under k-Lin, with a security loss that depends on the number of ciphertexts Q (via a hybrid argument) but independently of the number of PCA queries [11, 1].

Following the "randomized Naor-Reingold" paradigm introduced by Chen and Wee on tightly secure IBE [10], our starting point is (1), where we replace $\mathbf{k}_\tau = \mathbf{k}_0 + \tau \mathbf{k}_1$ with

$$\mathbf{k}_\tau = \sum_{j=1}^{\lambda} \mathbf{k}_{j,\tau_j}$$

and $\mathsf{pk} := ([\mathbf{M}], [\mathbf{M}^\top \mathbf{k}_{j,b}]_{j=1,\ldots,\lambda, b=0,1})$, where $(\tau_1, \ldots, \tau_\lambda)$ denotes the binary representation of the tag $\tau \in \{0,1\}^\lambda$.

Following [10], we want to analyze this construction by a sequence of games in which we first replace $[\mathbf{y}]$ in the challenge ciphertexts by uniformly random group elements via random self-reducibility of MDDH (k-Lin), and then incrementally replace $\mathbf{k}_\tau$ in both the challenge ciphertexts and in the PCA oracle by $\mathbf{k}_\tau + \mathbf{m}^\perp \mathsf{RF}(\tau)$, where RF is a truly random function and $\mathbf{m}^\perp$ is a random element from the kernel of $\mathbf{M}$, i.e., $\mathbf{M}^\top \mathbf{m}^\perp = 0$. Concretely, in Game i, we will replace $\mathbf{k}_\tau$ with $\mathbf{k}_\tau + \mathbf{m}^\perp \mathsf{RF}_i(\tau)$ where RF_i is a random function on $\{0,1\}^i$ applied to the i-bit prefix of τ. We proceed to outline the two main ideas needed to carry out this transition. Looking ahead, note that once we reach Game λ, we would have replaced $\mathbf{k}_\tau$ with $\mathbf{k}_\tau + \mathbf{m}^\perp \mathsf{RF}(\tau)$, upon which security follows from a straight-forward information-theoretic argument (and the fact that ciphertexts and decryption queries carry pairwise different τ).

First idea. First, we show how to transition from Game i to Game $i+1$, under the restriction that the adversary is only allowed to query the encryption oracle on tags whose $i+1$-st bit is 0; we show how to remove this unreasonable restriction later. Here, we rely on an *information-theoretic* argument similar to that of Cramer and Shoup to increase the entropy from RF_i to RF_{i+1}. This is in contrast to prior works which rely on a computational argument; note that the latter requires encoding secret keys as group elements and thus a pairing to carry out decryption.

More precisely, we pick a random function RF'_i on $\{0,1\}^i$, and implicitly define RF_{i+1} as follows:

$$\mathsf{RF}_{i+1}(\tau) = \begin{cases} \mathsf{RF}_i(\tau) & \text{if } \tau_{i+1} = 0 \\ \mathsf{RF}'_i(\tau) & \text{if } \tau_{i+1} = 1 \end{cases}$$

Observe all of the challenge ciphertexts leak no information about RF'_i or $\mathbf{k}_{i+1,1}$ since they all correspond to tags whose $i+1$-st bit is 0. To handle a PCA query $(\tau, [\mathbf{y}], [z])$, we proceed via a case analysis:

– if $\tau_{i+1} = 0$, then $\mathbf{k}_\tau + \mathsf{RF}_{i+1}(\tau) = \mathbf{k}_\tau + \mathsf{RF}_i(\tau)$ and the PCA oracle returns the same value in both Games i and $i+1$.
– if $\tau_{i+1} = 1$ and $\mathbf{y}$ lies in the span of $\mathbf{M}$, we have

$$\mathbf{y}^\top \mathbf{m}^\perp = 0 \Longrightarrow \mathbf{y}^\top(\mathbf{k}_\tau + \mathbf{m}^\perp \mathsf{RF}_i(\tau)) = \mathbf{y}^\top(\mathbf{k}_\tau + \mathbf{m}^\perp \mathsf{RF}_{i+1}(\tau)),$$

and again the PCA oracle returns the same value in both Games i and $i+1$.
– if $\tau_{i+1} = 1$ and $\mathbf{y}$ lies outside the span of $\mathbf{M}$, then $\mathbf{y}^\top \mathbf{k}_{i+1,1}$ is uniformly random given $\mathbf{M}, \mathbf{M}^\top \mathbf{k}_{i+1,1}$. (Here, we crucially use that the adversary does not query encryptions with $\tau_{i+1} = 1$, which ensures that the challenge ciphertexts do not leak additional information about $\mathbf{k}_{i+1,1}$.) This means that $\mathbf{y}^\top \mathbf{k}_\tau$ is uniformly random from the adversary's view-point, and therefore the PCA oracle will reject with high probability in both Games i and $i+1$. (At this point, we crucially rely on the fact that the PCA oracle only outputs a *single* check bit and not all of $\mathbf{k}_\tau + \mathsf{RF}(\tau)$.)

Via a hybrid argument, we may deduce that the distinguishing advantage between Games i and $i+1$ is at most Q/q where Q is the number of PCA queries.

Second idea. Next, we remove the restriction on the encryption queries using an idea of Hofheinz, Koch and Striecks [19] for tightly-secure IBE in the multi-ciphertext setting, and its instantiation in prime-order groups [16]. The idea is to create two "independent copies" of $(\mathbf{m}^\perp, \mathsf{RF}_i)$; we use one to handle encryption queries on tags whose $i+1$-st bit is 0, and the other to handle those whose $i+1$-st bit is 1. We call these two copies $(\mathbf{M}_0^*, \mathsf{RF}_i^{(0)})$ and $(\mathbf{M}_1^*, \mathsf{RF}_i^{(1)})$, where $\mathbf{M}^\top \mathbf{M}_0^* = \mathbf{M}^\top \mathbf{M}_1^* = \mathbf{0}$.

Concretely, we replace $\mathbf{M} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{(k+1)\times k}$ with $\mathbf{M} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k\times k}$. We decompose $\mathbb{Z}_q^{3k}$ into the span of the respective matrices $\mathbf{M}, \mathbf{M}_0, \mathbf{M}_1$, and we will also decompose the span of $\mathbf{M}^\perp \in \mathbb{Z}_q^{3k\times 2k}$ into that of $\mathbf{M}_0^*, \mathbf{M}_1^*$. Similarly, we decompose $\mathbf{M}^\perp \mathsf{RF}_i(\tau)$ into $\mathbf{M}_0^* \mathsf{RF}_i^{(0)}(\tau) + \mathbf{M}_1^* \mathsf{RF}_i^{(1)}(\tau)$. We then refine the

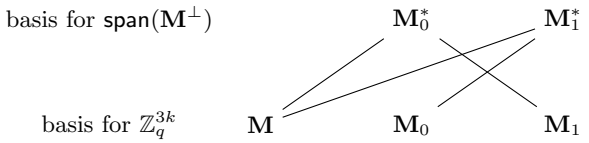

Fig. 1. Solid lines mean orthogonal, that is: $\mathbf{M}^\top \mathbf{M}_0^* = \mathbf{M}_1^\top \mathbf{M}_0^* = \mathbf{0} = \mathbf{M}^\top \mathbf{M}_1^* = \mathbf{M}_0^\top \mathbf{M}_1^*$.

prior transition from Games i to $i+1$ as follows:

– Game $i.0$ (= Game i): pick $\mathbf{y} \leftarrow \mathbb{Z}_q^{3k}$ for ciphertexts, and replace $\mathbf{k}_\tau$ with $\mathbf{k}_\tau + \mathbf{M}_0^* \mathsf{RF}_i^{(0)}(\tau) + \mathbf{M}_1^* \mathsf{RF}_i^{(1)}(\tau)$;
– Game $i.1$: replace $\mathbf{y} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k}$ with $\mathbf{y} \leftarrow_{\mathrm{R}} \mathsf{span}(\mathbf{M}, \mathbf{M}_{\tau_{i+1}})$;
– Game $i.2$: replace $\mathsf{RF}_i^{(0)}(\tau)$ with $\mathsf{RF}_{i+1}^{(0)}(\tau)$;
– Game $i.3$: replace $\mathsf{RF}_i^{(1)}(\tau)$ with $\mathsf{RF}_{i+1}^{(1)}(\tau)$;
– Game $i.4$ (= Game $i+1$): replace $\mathbf{y} \leftarrow_{\mathrm{R}} \mathsf{span}(\mathbf{M}, \mathbf{M}_{\tau_{i+1}})$ with $\mathbf{y} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k}$.

For the transition from Game $i.0$ to Game $i.1$, we rely on the fact that the uniform distributions over $\mathbb{Z}_q^{3k}$ and $\mathsf{span}(\mathbf{M}, \mathbf{M}_{\tau_{i+1}})$ encoded in the group are computationally indistinguishable, even given a

random basis for $\mathsf{span}(\mathbf{M}^\perp)$ (in the clear). This extends to the setting with multiple samples, with a tight reduction to the $\mathcal{D}_k$-MDDH Assumption independent of the number of samples.

For the transition from Game $i.1$ to $i.2$, we rely on an information-theoretic argument like the one we just outlined, replacing $\mathsf{span}(\mathbf{M})$ with $\mathsf{span}(\mathbf{M}, \mathbf{M}_1)$ and $\mathbf{M}^\perp$ with $\mathbf{M}_0^*$ in the case analysis. In particular, we will exploit the fact that if $\mathbf{y}$ lies outside $\mathsf{span}(\mathbf{M}, \mathbf{M}_1)$, then $\mathbf{y}^\top \mathbf{k}_{i+1,1}$ is uniformly random even given $\mathbf{M}, \mathbf{M}\mathbf{k}_{i+1,1}, \mathbf{M}_1, \mathbf{M}_1\mathbf{k}_{i+1,1}$. The transition from Game $i.2$ to $i.3$ is completely analogous.

From PCA to CCA. Using standard techniques from [11, 23, 21, 8, 4], we could transform our basic tag-based PCA-secure scheme into a "full-fledged" CCA-secure encryption scheme by adding another hash proof system (or an authenticated symmetric encryption scheme) and a one-time signature scheme. However, this would incur an additional overhead of several group elements in the ciphertext. Instead, we show how to directly modify our tag-based PCA-secure scheme to obtain a more efficient CCA-secure scheme with the minimal additional overhead of a single symmetric-key authenticated encryption. In particular, the overall ciphertext overhead in our tightly CCA-secure encryption scheme is merely *one* group element more than that for the best known non-tight schemes [23, 18].

To encrypt a message M in the CCA-secure encryption scheme, we will (i) pick a random $\mathbf{y}$ as in the tag-based PCA scheme, (ii) derive a tag τ from $\mathbf{y}$, (iii) encrypt M using a one-time authenticated encryption under the KEM key $[\mathbf{y}^\top \mathbf{k}_\tau]$. The naive approach is to derive the tag τ by hashing $[\mathbf{y}] \in \mathbb{G}^{3k}$, as in [23]. However, this creates a circularity in Game $i.1$ where the distribution of $[\mathbf{y}]$ depends on the tag. Instead, we will derive the tag τ by hashing $[\overline{\mathbf{y}}] \in \mathbb{G}^k$, where $\overline{\mathbf{y}} \in \mathbb{Z}_q^k$ are the top k entries of $\mathbf{y} \in \mathbb{Z}_q^{3k}$. We then modify $\mathbf{M}_0, \mathbf{M}_1$ so that the top k rows of both matrices are zero, which avoids the circularity issue. In the proof of security, we will also rely on the fact that for any $\mathbf{y}_0, \mathbf{y}_1 \in \mathbb{Z}_q^{3k}$, if $\overline{\mathbf{y}}_0 = \overline{\mathbf{y}}_1$ and $\mathbf{y}_0 \in \mathsf{span}(\mathbf{M})$, then either $\mathbf{y}_0 = \mathbf{y}_1$ or $\mathbf{y}_1 \notin \mathsf{span}(\mathbf{M})$. This allows us to deduce that if the adversary queries the CCA oracle on a ciphertext which shares the same tag as some challenge ciphertext, then the CCA oracle will reject with overwhelming probability.

Alternative view-point. Our construction can also be viewed as applying the BCHK IBE→PKE transform [8] to the scheme from [19], and then writing the exponents of the secret keys in the clear, thereby avoiding the pairing. This means that we can no longer apply a computational assumption and the randomized Naor-Reingold argument to the secret key space. Indeed, we replace this with an information-theoretic Cramer-Shoup-like argument as outlined above.

Prior approaches. Several approaches to construct tightly CCA-secure PKE schemes exist: first, the schemes of [17, 2, 3, 25, 24, 26] construct a tightly secure NIZK scheme from a tightly secure signature scheme, and then use the tightly secure NIZK in a CCA-secure PKE scheme following the Naor-Yung double encryption paradigm [30, 13]. Since these approaches build on the public verifiability of the used NIZK scheme (in order to faithfully simulate a decryption oracle), their reliance on a pairing seems inherent.

Next, the works of [10, 7, 19, 5, 16] used a (Naor-Reingold-based) MAC instead of a signature scheme to design tightly secure IBE schemes. Those IBE schemes can then be converted (using the BCHK transformation [8]) into tightly CCA-secure PKE schemes. However, the derived PKE schemes still rely on pairings, since the original IBE schemes do (and the BCHK does not remove the reliance on pairings).

Reference	$\|\mathsf{pk}\|$	$\|\mathsf{ct}\| - \|m\|$	security loss	assumption	pairing
CS98 [11]	$\mathcal{O}(1)$	3	$\mathcal{O}(Q)$	DDH	no
KD04, HK07 [23, 18]	$\mathcal{O}(1)$	2	$\mathcal{O}(Q)$	DDH	no
HJ12 [17]	$O(1)$	$O(\lambda)$	$\mathcal{O}(1)$	2-Lin	yes
LPJY15 [24, 26]	$\mathcal{O}(\lambda)$	47	$\mathcal{O}(\lambda)$	2-Lin	yes
AHY15 [5]	$\mathcal{O}(\lambda)$	12	$\mathcal{O}(\lambda)$	2-Lin	yes
GCDCT15 [16]	$\mathcal{O}(\lambda)$	10 (resp. $6k+4$)	$\mathcal{O}(\lambda)$	SXDH (resp. k-Lin)	yes
Ours §4	$\mathcal{O}(\lambda)$	3 (resp. $3k$)	$\mathcal{O}(\lambda)$	DDH (resp. k-Lin)	no

Fig. 2. Comparison amongst CCA-secure encryption schemes, where Q is the number of ciphertexts, $|\mathsf{pk}|$ denotes the size (i.e the number of groups elements, or exponent of group elements) of the public key, and $|\mathsf{ct}|-|m|$ denotes the ciphertext overhead, ignoring smaller contributions from symmetric-key encryption. We omit [19] from this table since we only focus on prime-order groups here.

In contrast, our approach directly fuses a Naor-Reingold-like randomization argument with the encryption process. We are able to do so since we substitute a computational randomization argument (as used in the latter line of works) with an information-theoretic one, as described above. Hence, we can apply that argument to *exponents* rather than group elements. This enables us to trade pairing operations for exponentiations in our scheme.

Efficiency comparison with non-tightly secure schemes. We finally mention that our DDH-based scheme compares favorably even with the most efficient (non-tightly) CCA-secure DDH-based encryption schemes [23, 18]. To make things concrete, assume $\lambda = 80$ and a setting with $Q_{\mathsf{enc}} = Q_{\mathsf{dec}} = 2^{30}$. The best known reductions for the schemes of [23, 18] lose a factor of $Q_{\mathsf{enc}} = 2^{30}$, whereas our scheme loses a factor of about $4\lambda \leq 2^9$. Hence, the group size for [23, 18] should be at least $2^{2\cdot(80+30)} = 2^{220}$ compared to $2^{2\cdot(80+9)} = 2^{178}$ in our case. Thus, the ciphertext overhead (ignoring the symmetric encryption part) in our scheme is $3 \cdot 178 = 534$ bits, which is close to $2 \cdot 220 = 440$ bits with [23, 18].[5]

Perhaps even more interestingly, we can compare computational efficiency of encryption in this scenario. For simplicitly, we only count exponentiations and assume a naive square-and-multiply-based exponentiation with no further multi-exponentiation optimizations.[6] Encryption in [23, 18] takes about 3.5 exponentiations (where we count an exponentiation with a $(\lambda + \log_2(Q_{\mathsf{enc}} + Q_{\mathsf{dec}}))$-bit hash value[7] as 0.5 exponentiations). In our scheme, we have about 4.67 exponentiations, where we count the computation of $[\mathbf{M}^\top \mathbf{k}_\tau]$ – which consists of 2λ multiplications – as 0.67 exponentiations.) Since exponentiation (under our assumptions) takes time cubic in the bitlength, we get that encryption with our scheme is actually about 29% *less expensive* than with [23, 18].

However, of course we should also note that public and secret key in our scheme are significantly larger (e.g., $4\lambda + 3 = 323$ group elements in pk) than with [23, 18] (4 group elements in pk).

Extension: NIZK arguments. We also obtain tightly simulation-sound non-interactive zero-knowledge (NIZK) arguments from our encryption scheme in a semi-generic way.

Let us start with any designated-verifier quasi-adaptive NIZK (short: DVQANIZK) argument system Π for a given language. Recall that in a designated-verifier NIZK, proofs can only be

[5] In this calculation, we do not consider the symmetric authenticated encryption of the actual plaintext (and a corresponding MAC value), which is the same with [23, 18] and our scheme.

[6] Here, optimizations would improve the schemes of [23, 18] and ours similarly, since the schemes are very similar.

[7] It is possible to prove the security of [23, 18] using a *target*-collision-resistant hash function, such that $|\tau| = \lambda$. However, in the multi-user setting, a hybrid argument is required, such that the output size of the hash function will have to be increased to at least $|\tau| = \lambda + \log_2(Q_{\mathsf{enc}} + Q_{\mathsf{dec}})$.

verified with a secret verification key, and soundness only holds against adversaries who do not know that key. Furthermore, quasi-adaptivity means that the language has to be fixed at setup time of the scheme. Let Π_{PKE} be the variant of Π in which proofs are encrypted using a CCA-secure PKE scheme PKE. Public and secret key of PKE are of course made part of CRS and verification key, respectively. Observe that Π_{PKE} enjoys simulation-soundness, assuming that simulated proofs are simply encryptions of random plaintexts. Indeed, the CCA security of PKE guarantees that authentic Π_{PKE}-proofs can be substituted with simulated ones, while being able to verify (using a decryption oracle) a purported Π_{PKE}-proof generated by an adversary. Furthermore, if PKE is tightly secure, then so is Π_{PKE}.

When using a hash proof system for Π and our encryption scheme for PKE, this immediately yields a tightly simulation-sound DVQANIZK for linear languages (i.e., languages of the form $\{[\mathbf{Mx}] \mid \mathbf{x} \in \mathbb{Z}_q^t\}$ for some matrix $\mathbf{M} \in \mathbb{Z}_q^{n\times t}$ with $t < n$) that does not require pairings. We stress that our DVQANIZK is tightly secure in a setting with many simulated proofs and many adversarial verification queries.

Using the semi-generic transformation of [22], we can then derive a tightly simulation-sound QANIZK proof system (with public verification), that however relies on pairings. We note that the transformation of [22] only requires a DVQANIZK that is secure against a single adversarial verification query, since the pairing enables the public verifiability of proofs. Hence, we can first optimize and trim down our DVQANIZK (such that only a single adversarial verification query is supported), and then apply the transformation. This yields a QANIZK with particularly compact proofs. See Figure 3 for a comparison with relevant existing proof systems.

Reference	type	$\lvert\mathsf{crs}\rvert$	$\lvert\pi\rvert$	sec. loss	assumption	pairing
CCS09 [9]	NIZK	$\mathcal{O}(1)$	$2n+6t+52$	$\mathcal{O}(Q_{\mathsf{sim}})$	2-Lin	yes
HJ12 [17]	NIZK	$\mathcal{O}(1)$	$\gg 500$	$\mathcal{O}(1)$	2-Lin	yes
LPJY14 [25]	QANIZK	$\mathcal{O}(n+\lambda)$	20	$\mathcal{O}(Q_{\mathsf{sim}})$	2-Lin	yes
KW15 [22]	QANIZK	$\mathcal{O}(kn)$	$2k+2$	$\mathcal{O}(Q_{\mathsf{sim}})$	k-Lin	yes
LPJY15 [27]	QANIZK	$\mathcal{O}(n+\lambda)$	42	$\mathcal{O}(\lambda)$	2-Lin	yes
Ours §6.2	DVQANIZK	$\mathcal{O}(t+k\lambda)$	$3k+1$	$\mathcal{O}(\lambda)$	k-Lin	no
Ours §6.3	QANIZK	$\mathcal{O}(k^2\lambda+kn)$	$2k+1$	$\mathcal{O}(\lambda)$	k-Lin	yes

Fig. 3. (DV)QANIZK schemes for subspaces of $\mathbb{G}^n$ of dimension $t < n$. $\lvert\mathsf{crs}\rvert$ and $\lvert\pi\rvert$ denote the size (in group elements) of the CRS and of proofs. Q_{sim} is the number of simulated proofs in the simulation-soundness experiment. The scheme from [22] (as well as our own schemes) can also be generalized to matrix assumptions [14], at the cost of a larger CRS.

Roadmap. We recall some notation and basic definitions (including those concerning our algebraic setting and for tightly secure encryption) in Section 2. Section 3 presents our basic PCA-secure encryption scheme and represents the core of our results. In Section 4, we present our optimized CCA-secure PKE scheme. Our NIZK-related applications are presented in Section 6.

2 Preliminaries

2.1 Notations

If $\mathbf{x} \in \mathcal{B}^n$, then $|\mathbf{x}|$ denotes the length n of the vector. Further, $x \leftarrow_{\mathrm{R}} \mathcal{B}$ denotes the process of sampling an element x from set $\mathcal{B}$ uniformly at random. For any bit string $\tau \in \{0,1\}^*$, we denote

by τ_i the i'th bit of τ. We denote by λ the security parameter, and by $\mathsf{negl}(\cdot)$ any negligible function of λ. For all matrix $\mathbf{A} \in \mathbb{Z}_q^{\ell \times k}$ with $\ell > k$, $\overline{\mathbf{A}} \in \mathbb{Z}_q^{k \times k}$ denotes the upper square matrix of $\mathbf{A}$ and $\underline{\mathbf{A}} \in \mathbb{Z}_q^{\ell-k \times k}$ denotes the lower $\ell - k$ rows of $\mathbf{A}$. With $\mathsf{span}(\mathbf{A}) := \{\mathbf{Ar} \mid \mathbf{r} \in \mathbb{Z}_q^k\} \subset \mathbb{Z}_q^\ell$, we denote the span of $\mathbf{A}$.

2.2 Collision resistant hashing

A hash function generator is a PPT algorithm $\mathcal{H}$ that, on input 1^λ , outputs an efficiently computable function $\mathsf{H} : \{0,1\}^* \to \{0,1\}^\lambda$.

Definition 1 (Collision Resistance). *We say that a hash function generator $\mathcal{H}$ outputs collision-resistant functions H if for all PPT adversaries $\mathcal{A}$,*

$$\mathbf{Adv}^{\mathrm{cr}}_{\mathcal{H}}(\mathcal{A}) := \Pr[x \neq x' \wedge \mathsf{H}(x) = \mathsf{H}(x') | \mathsf{H} \leftarrow_{\mathrm{R}} \mathcal{H}(1^\lambda), (x, x') \leftarrow \mathcal{A}(1^\lambda, \mathsf{H})] = \mathsf{negl}(\lambda).$$

2.3 Prime-order groups

Let GGen be a probabilistic polynomial time (PPT) algorithm that on input 1^λ returns a description $\mathcal{G} = (\mathbb{G}, q, P)$ of an additive cyclic group $\mathbb{G}$ of order q for a λ-bit prime q, whose generator is P.

We use implicit representation of group elements as introduced in [14]. For $a \in \mathbb{Z}_q$, define $[a] = aP \in \mathbb{G}$ as the *implicit representation* of a in $\mathbb{G}$. More generally, for a matrix $\mathbf{A} = (a_{ij}) \in \mathbb{Z}_q^{n \times m}$ we define $[\mathbf{A}]$ as the implicit representation of $\mathbf{A}$ in $\mathbb{G}$:

$$[\mathbf{A}] := \begin{pmatrix} a_{11}P \ ... \ a_{1m}P \\ \\ a_{n1}P \ ... \ a_{nm}P \end{pmatrix} \in \mathbb{G}^{n \times m}$$

We will always use this implicit notation of elements in $\mathbb{G}$, i.e., we let $[a] \in \mathbb{G}$ be an element in $\mathbb{G}$. Note that from $[a] \in \mathbb{G}$ it is generally hard to compute the value a (discrete logarithm problem in $\mathbb{G}$). Obviously, given $[a], [b] \in \mathbb{G}$ and a scalar $x \in \mathbb{Z}_q$, one can efficiently compute $[ax] \in \mathbb{G}$ and $[a + b] \in \mathbb{G}$.

2.4 Matrix Diffie-Hellman Assumption

We recall the definitions of the Matrix Decision Diffie-Hellman (MDDH) Assumption [14].

Definition 2 (Matrix Distribution). *Let $k, \ell \in \mathbb{N}$, with $\ell > k$. We call $\mathcal{D}_{\ell,k}$ a matrix distribution if it outputs matrices in $\mathbb{Z}_q^{\ell \times k}$ of full rank k in polynomial time. We write $\mathcal{D}_k := \mathcal{D}_{k+1,k}$.*

Without loss of generality, we assume the first k rows of $\mathbf{A} \leftarrow_{\mathrm{R}} \mathcal{D}_{\ell,k}$ form an invertible matrix. The $\mathcal{D}_{\ell,k}$-Matrix Diffie-Hellman problem is to distinguish the two distributions $([\mathbf{A}], [\mathbf{Aw}])$ and $([\mathbf{A}], [\mathbf{u}])$ where $\mathbf{A} \leftarrow_{\mathrm{R}} \mathcal{D}_{\ell,k}$, $\mathbf{w} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$ and $\mathbf{u} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^\ell$.

Definition 3 ($\mathcal{D}_{\ell,k}$-Matrix Diffie-Hellman Assumption $\mathcal{D}_{\ell,k}$-MDDH). *Let $\mathcal{D}_{\ell,k}$ be a matrix distribution. We say that the $\mathcal{D}_{\ell,k}$-Matrix Diffie-Hellman ($\mathcal{D}_{\ell,k}$-MDDH) Assumption holds relative to GGen if for all PPT adversaries $\mathcal{A}$,*

$$\mathbf{Adv}^{\mathrm{mddh}}_{\mathcal{D}_{\ell,k},\mathsf{GGen}}(\mathcal{A}) := |\Pr[\mathcal{A}(\mathcal{G}, [\mathbf{A}], [\mathbf{Aw}]) = 1] - \Pr[\mathcal{A}(\mathcal{G}, [\mathbf{A}], [\mathbf{u}]) = 1]| = \mathsf{negl}(\lambda),$$

where the probability is taken over $\mathcal{G} \leftarrow_{\mathrm{R}} \mathsf{GGen}(1^\lambda)$, $\mathbf{A} \leftarrow_{\mathrm{R}} \mathcal{D}_k, \mathbf{w} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k, \mathbf{u} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^\ell$.

For each $k \geq 1$, [14] specifies distributions $\mathcal{L}_k$, $\mathcal{SC}_k$, $\mathcal{C}_k$ (and others) over $\mathbb{Z}_q^{(k+1)\times k}$ such that the corresponding $\mathcal{D}_k$-MDDH assumptions are generically secure in bilinear groups and form a hierarchy of increasingly weaker assumptions. $\mathcal{L}_k$-MDDH is the well known k-Linear Assumption k-Lin with 1-Lin = DDH. In this work we are mostly interested in the uniform matrix distribution $\mathcal{U}_{\ell,k}$.

Definition 4 (Uniform distribution). *Let $\ell, k \in \mathbb{N}$, with $\ell > k$. We denote by $\mathcal{U}_{\ell,k}$ the uniform distribution over all full-rank $\ell \times k$ matrices over $\mathbb{Z}_q$. Let $\mathcal{U}_k := \mathcal{U}_{k+1,k}$.*

Lemma 1 ($\mathcal{U}_k$-MDDH $\Leftrightarrow$ $\mathcal{U}_{\ell,k}$-MDDH). *Let $\ell, k \in \mathbb{N}$, with $\ell > k$. For any PPT adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ (and vice versa) such that $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ and $\mathbf{Adv}^{\mathrm{mddh}}_{\mathcal{U}_{\ell,k},\mathsf{GGen}}(\mathcal{A}) = \mathbf{Adv}^{\mathrm{mddh}}_{\mathcal{U}_k,\mathsf{GGen}}(\mathcal{B})$.*

Proof. This follows from the simple fact that a $\mathcal{U}_{\ell,k}$-MDDH instance $([\mathbf{A}], [\mathbf{z}])$ can be transformed into an $\mathcal{U}_k$-MDDH instance $([\mathbf{A}'] = [\mathbf{TA}], [\mathbf{z}'] = [\mathbf{Tz}])$ for a random $(k+1) \times \ell$ matrix $\mathbf{T}$. If $\mathbf{z} = \mathbf{Aw}$, then $\mathbf{z}' = \mathbf{TAw} = \mathbf{A}'\mathbf{w}$; if $\mathbf{z}$ is uniform, so is $\mathbf{z}'$. Similarly, a $\mathcal{U}_k$-MDDH instance $([\mathbf{A}'], [\mathbf{z}'])$ can be transformed into an $\mathcal{U}_{\ell,k}$-MDDH instance $([\mathbf{A}] = [\mathbf{T}'\mathbf{A}'], [\mathbf{z}] = [\mathbf{T}'\mathbf{z}'])$ for a random $\ell \times (k+1)$ matrix $\mathbf{T}'$. □

Among all possible matrix distributions $\mathcal{D}_{\ell,k}$, the uniform matrix distribution $\mathcal{U}_k$ is the hardest possible instance, so in particular k-Lin $\Rightarrow \mathcal{U}_k$-MDDH.

Lemma 2 ($\mathcal{D}_{\ell,k}$-MDDH $\Rightarrow \mathcal{U}_k$-MDDH, [14]). *Let $\mathcal{D}_{\ell,k}$ be a matrix distribution. For any PPT adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ and $\mathbf{Adv}^{\mathrm{mddh}}_{\mathcal{D}_{\ell,k},\mathsf{GGen}}(\mathcal{A}) = \mathbf{Adv}^{\mathrm{mddh}}_{\mathcal{U}_k,\mathsf{GGen}}(\mathcal{B})$.*

Let $Q \geq 1$. For $\mathbf{W} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{k\times Q}, \mathbf{U} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{\ell\times Q}$, we consider the Q-fold $\mathcal{D}_{\ell,k}$-MDDH Assumption which consists in distinguishing the distributions $([\mathbf{A}], [\mathbf{AW}])$ from $([\mathbf{A}], [\mathbf{U}])$. That is, a challenge for the Q-fold $\mathcal{D}_{\ell,k}$-MDDH Assumption consists of Q independent challenges of the $\mathcal{D}_{\ell,k}$-MDDH Assumption (with the same $\mathbf{A}$ but different randomness $\mathbf{w}$). In [14] it is shown that the two problems are equivalent, where (for $Q \geq \ell - k$) the reduction loses a factor $\ell - k$. In combination with Lemma 1 we obtain the following tighter version for the special case of $\mathcal{D}_{\ell,k} = \mathcal{U}_{\ell,k}$.

Lemma 3 (Random self-reducibility of $\mathcal{U}_{\ell,k}$-MDDH, [14]). *Let $\ell, k, Q \in \mathbb{N}$ with $\ell > k$. For any PPT adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A}) + Q \cdot \mathsf{poly}(\lambda)$ with $\mathsf{poly}(\lambda)$ independent of $\mathbf{T}(\mathcal{A})$, and*

$$\mathbf{Adv}^{\text{Q-mddh}}_{\mathcal{U}_{\ell,k},\mathsf{GGen}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{mddh}}_{\mathcal{U}_{\ell,k},\mathsf{GGen}}(\mathcal{B}) + \frac{1}{q-1}$$

where $\mathbf{Adv}^{\text{Q-mddh}}_{\mathcal{U}_{\ell,k},\mathsf{GGen}}(\mathcal{B}) := |\Pr[\mathcal{B}(\mathcal{G}, [\mathbf{A}], [\mathbf{AW}]) = 1] - \Pr[\mathcal{B}(\mathcal{G}, [\mathbf{A}], [\mathbf{U}]) = 1]|$ and the probability is over $\mathcal{G} \leftarrow_{\mathrm{R}} \mathsf{GGen}(1^\lambda)$, $\mathbf{A} \leftarrow_{\mathrm{R}} \mathcal{U}_{\ell,k}, \mathbf{W} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{k\times Q}, \mathbf{U} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{\ell\times Q}$.

2.5 Public-Key Encryption

Definition 5 (PKE). *A* Public-Key Encryption *(PKE) consists of three PPT algorithms* $\mathsf{PKE} = (\mathsf{Param}_{\mathsf{PKE}}, \mathsf{Gen}_{\mathsf{PKE}}, \mathsf{Enc}_{\mathsf{PKE}}, \mathsf{Dec}_{\mathsf{PKE}})$*:*

- *The probabilistic key generation algorithm $\mathsf{Gen}_{\mathsf{PKE}}(1^\lambda)$ generates a pair of public and secret keys $(\mathsf{pk}, \mathsf{sk})$.*
- *The probabilistic encryption algorithm $\mathsf{Enc}_{\mathsf{PKE}}(\mathsf{pk}, M)$ returns a ciphertext ct.*

– *The deterministic decryption algorithm* $\mathsf{Dec}_{\mathsf{PKE}}(\mathsf{pk}, \mathsf{sk}, \mathsf{ct})$ *returns a message* M *or* $\perp$*, where* $\perp$ *is a special rejection symbol.*

We define the following properties:

Perfect correctness. *For all* λ*, we have*

$$\Pr\left[\mathsf{Dec}_{\mathsf{PKE}}(\mathsf{pk}, \mathsf{sk}, \mathsf{ct}) = M \,\middle|\, \begin{array}{l}(\mathsf{pk}, \mathsf{sk}) \leftarrow_{\mathrm{R}} \mathsf{Gen}_{\mathsf{PKE}}(1^\lambda); \\ \mathsf{ct} \leftarrow_{\mathrm{R}} \mathsf{Enc}_{\mathsf{PKE}}(\mathsf{pk}, M)\end{array}\right] = 1.$$

Multi-ciphertext CCA security [6]. *For any adversary* $\mathcal{A}$*, we define*

$$\mathbf{Adv}^{\mathsf{ind\text{-}cca}}_{\mathsf{PKE}}(\mathcal{A}) := \left|\Pr\left[b = b' \,\middle|\, b' \leftarrow \mathcal{A}^{\mathsf{Setup},\mathsf{DecO}(\cdot),\mathsf{EncO}(\cdot,\cdot)}(1^\lambda)\right] - 1/2\right|$$

where:

– Setup *sets* $\mathcal{C}_{\mathsf{enc}} := \emptyset$*, samples* $(\mathsf{pk}, \mathsf{sk}) \leftarrow_{\mathrm{R}} \mathsf{Gen}_{\mathsf{KEM}}(1^\lambda)$ *and* $b \leftarrow_{\mathrm{R}} \{0, 1\}$*, and returns* pk. Setup *must be called once at the beginning of the game.*

– $\mathsf{DecO}(\mathsf{ct})$ *returns* $\mathsf{Dec}_{\mathsf{PKE}}(\mathsf{pk}, \mathsf{sk}, \mathsf{ct})$ *if* $\mathsf{ct} \notin \mathcal{C}_{\mathsf{enc}}$*,* $\perp$ *otherwise.*

– *If* M_0 *and* M_1 *are two messages of equal length,* $\mathsf{EncO}(M_0, M_1)$ *returns* $\mathsf{Enc}_{\mathsf{PKE}}(\mathsf{pk}, M_b)$ *and sets* $\mathcal{C}_{\mathsf{enc}} := \mathcal{C}_{\mathsf{enc}} \cup \{\mathsf{ct}\}$.

We say PKE *is* IND-CCA *secure if for all* PPT *adversaries* $\mathcal{A}$*, the advantage* $\mathbf{Adv}^{\mathsf{ind\text{-}cca}}_{\mathsf{PKE}}(\mathcal{A})$ *is a negligible function of* λ*.*

2.6 Key-Encapsulation Mechanism

Definition 6 (Tag-based KEM). *A tag-based* Key-Encapsulation Mechanism *(KEM) consists of three PPT algorithms* $\mathsf{KEM} = (\mathsf{Gen}_{\mathsf{KEM}}, \mathsf{Enc}_{\mathsf{KEM}}, \mathsf{Dec}_{\mathsf{KEM}})$*:*

– *The probabilistic key generation algorithm* $\mathsf{Gen}_{\mathsf{KEM}}(1^\lambda)$ *generates a pair of public and secret keys* $(\mathsf{pk}, \mathsf{sk})$.

– *The probabilistic encryption algorithm* $\mathsf{Enc}_{\mathsf{KEM}}(\mathsf{pk}, \tau)$ *returns a pair* (K, C) *where* K *is a uniformly distributed symmetric key in* $\mathcal{K}$ *and* C *is a ciphertext, with respect to the tag* $\tau \in \mathcal{T}$*.*

– *The deterministic decryption algorithm* $\mathsf{Dec}_{\mathsf{KEM}}(\mathsf{pk}, \mathsf{sk}, \tau, C)$ *returns a key* $K \in \mathcal{K}$*.*

We define the following properties:

Perfect correctness. *For all* λ*, for all tags* $\tau \in \mathcal{T}$*, we have*

$$\Pr\left[\mathsf{Dec}_{\mathsf{KEM}}(\mathsf{pk}, \mathsf{sk}, \tau, C) = K \,\middle|\, \begin{array}{l}(\mathsf{pk}, \mathsf{sk}) \leftarrow_{\mathrm{R}} \mathsf{Gen}_{\mathsf{KEM}}(1^\lambda); \\ (K, C) \leftarrow_{\mathrm{R}} \mathsf{Enc}_{\mathsf{KEM}}(\mathsf{pk}, \tau)\end{array}\right] = 1.$$

Multi-ciphertext PCA security [31]. *For any adversary* $\mathcal{A}$*, we define*

$$\mathbf{Adv}^{\mathsf{ind\text{-}pca}}_{\mathsf{KEM}}(\mathcal{A}) := \left|\Pr\left[b = b' \,\middle|\, b' \leftarrow \mathcal{A}^{\mathsf{Setup},\mathsf{DecO}(\cdot,\cdot,\cdot),\mathsf{EncO}(\cdot)}(1^\lambda)\right] - 1/2\right|$$

where:

– Setup *sets* $\mathcal{T}_{\mathsf{enc}} = \mathcal{T}_{\mathsf{dec}} := \emptyset$*, samples* $(\mathsf{pk}, \mathsf{sk}) \leftarrow_{\mathrm{R}} \mathsf{Gen}_{\mathsf{KEM}}(1^\lambda)$*, picks* $b \leftarrow_{\mathrm{R}} \{0, 1\}$*, and returns* pk. Setup *is called once at the beginning of the game.*

– *The decryption oracle* $\mathsf{DecO}(\tau, C, \widehat{K})$ *computes* $K := \mathsf{Dec}_{\mathsf{KEM}}(\mathsf{pk}, \mathsf{sk}, \tau, C)$*. It returns* 1 *if* $\widehat{K} = K \wedge \tau \notin \mathcal{T}_{\mathsf{enc}}$*,* 0 *otherwise. Then it sets* $\mathcal{T}_{\mathsf{dec}} := \mathcal{T}_{\mathsf{dec}} \cup \{\tau\}$.

– $\mathsf{EncO}(\tau)$ *computes* $(K, C) \leftarrow_{\mathrm{R}} \mathsf{Enc}_{\mathsf{KEM}}(\mathsf{pk}, \tau)$*, sets* $K_0 := K$ *and* $K_1 \leftarrow_{\mathrm{R}} \mathcal{K}$*. If* $\tau \notin \mathcal{T}_{\mathsf{dec}} \cup \mathcal{T}_{\mathsf{enc}}$*, it returns* (C, K_b)*, and sets* $\mathcal{T}_{\mathsf{enc}} := \mathcal{T}_{\mathsf{enc}} \cup \{\tau\}$*; otherwise it returns* $\perp$*.*

We say KEM *is* IND-PCA *secure if for all* PPT *adversaries* $\mathcal{A}$*, the advantage* $\mathbf{Adv}^{\mathsf{ind\text{-}pca}}_{\mathsf{KEM}}(\mathcal{A})$ *is a negligible function of* λ*.*

2.7 Authenticated Encryption

Definition 7 (AE [18]). *An* authenticated symmetric encryption *(AE) with message-space $\mathcal{M}$ and key-space $\mathcal{K}$ consists of two polynomial-time deterministic algorithms* $(\mathsf{Enc}_{\mathsf{AE}}, \mathsf{Dec}_{\mathsf{AE}})$*:*

- *The encryption algorithm* $\mathsf{Enc}_{\mathsf{AE}}(K, M)$ *generates C, encryption of the message M with the secret key K.*
- *The decryption algorithm* $\mathsf{Dec}_{\mathsf{AE}}(K, C)$*, returns a message M or $\perp$.*

We require that the algorithms satisfy the following properties:

Perfect correctness. *For all λ, for all $K \in \mathcal{K}$ and $M \in \mathcal{M}$, we have*

$$\mathsf{Dec}_{\mathsf{AE}}(K, \mathsf{Enc}_{\mathsf{AE}}(K, M)) = M.$$

One-time Privacy and Authenticity. *For any PPT adversary $\mathcal{A}$,*

$$\mathbf{Adv}^{\mathsf{ae\text{-}ot}}_{\mathsf{AE}}(\mathcal{A}) := \left| \Pr\left[b' = b \;\middle|\; \begin{matrix} K \leftarrow_{\mathrm{R}} \mathcal{K}; b \leftarrow_{\mathrm{R}} \{0,1\} \\ b' \leftarrow_{\mathrm{R}} \mathcal{A}^{\mathsf{ot\text{-}EncO}(\cdot,\cdot),\mathsf{ot\text{-}DecO}(\cdot)}(1^\lambda, \mathcal{M}, \mathcal{K}) \end{matrix} \right] - 1/2 \right|$$

is negligible, where $\mathsf{ot\text{-}EncO}(M_0, M_1)$*, on input two messages M_0 and M_1 of the same length,* $\mathsf{Enc}_{\mathsf{AE}}(K, M_b)$*, and* $\mathsf{ot\text{-}DecO}(\phi)$ *returns* $\mathsf{Dec}_{\mathsf{AE}}(K, \phi)$ *if $b = 0$, $\perp$ otherwise. $\mathcal{A}$ is allowed at most one call to each oracle* ot-EncO *and* ot-DecO*, and the query to* ot-DecO *must be different from the output of* ot-EncO*. $\mathcal{A}$ is also given the description of the key-space $\mathcal{K}$ as input.*

3 Multi-ciphertext PCA-secure KEM

In this section we describe a tag-based Key Encapsulation Mechanism $\mathsf{KEM}_{\mathsf{PCA}}$ that is IND-PCA-secure (see Definition 6).

For simplicity, we use the matrix distribution $\mathcal{U}_{3k,k}$ in our scheme in Figure 4, and prove it secure under the $\mathcal{U}_k$-MDDH Assumption ($\Leftrightarrow \mathcal{U}_{3k,k}$-MDDH Assumption, by Lemma 1), which in turn admits a tight reduction to the standard k-Lin Assumption. However, using a matrix distribution $\mathcal{D}_{3k,k}$ with more compact representation yields a more efficient scheme, secure under the $\mathcal{D}_{3k,k}$-MDDH Assumption (see Remark 1).

3.1 Our construction

Remark 1 (On the use of the $\mathcal{U}_k$-MDDH Assumption). In our scheme, we use a matrix distribution $\mathcal{U}_{3k,k}$ for the matrix $\mathbf{M}$, therefore proving security under the $\mathcal{U}_{3k,k}$-MDDH Assumption $\Leftrightarrow$ $\mathcal{U}_k$-MDDH Assumption (see Lemma 2). This is for simplicity of presentation. However, for efficiency, one may want to use an assumption with a more compact representation, such as the $\mathcal{CI}_{3k,k}$-MDDH Assumption [28] with representation size $2k$ instead of $3k^2$ for $\mathcal{U}_{3k,k}$.

3.2 Security proof

Theorem 1. *The tag-based Key Encapsulation Mechanism* $\mathsf{KEM}_{\mathsf{PCA}}$ *defined in Figure 4 has perfect correctness. Moreover, if the $\mathcal{U}_k$-MDDH Assumption holds in $\mathbb{G}$,* $\mathsf{KEM}_{\mathsf{PCA}}$ *is* IND-PCA *secure.*

$\underline{\mathsf{Gen}_{\mathsf{KEM}}(1^\lambda)}$:	$\underline{\mathsf{Enc}_{\mathsf{KEM}}(\mathsf{pk}, \tau)}$:
$\mathcal{G} \leftarrow_{\mathrm{R}} \mathsf{GGen}(1^\lambda)$; $\mathbf{M} \leftarrow_{\mathrm{R}} \mathcal{U}_{3k,k}$	$\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$; $C := [\mathbf{r}^\top \mathbf{M}^\top]$
$\mathbf{k}_{1,0}, \ldots, \mathbf{k}_{\lambda,1} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k}$	$\mathbf{k}_\tau := \sum_{j=1}^{\lambda} \mathbf{k}_{j,\tau_j}$
$\mathsf{pk} := \left(\mathcal{G}, [\mathbf{M}], \left([\mathbf{M}^\top \mathbf{k}_{j,\beta}]\right)_{1 \le j \le \lambda, 0 \le \beta \le 1}\right)$	$K := [\mathbf{r}^\top \cdot \mathbf{M}^\top \mathbf{k}_\tau]$
$\mathsf{sk} := (\mathbf{k}_{j,\beta})_{1 \le j \le \lambda, 0 \le \beta \le 1}$	Return $(C, K) \in \mathbb{G}^{1 \times 3k} \times \mathbb{G}$
Return $(\mathsf{pk}, \mathsf{sk})$	
	$\underline{\mathsf{Dec}_{\mathsf{KEM}}(\mathsf{pk}, \mathsf{sk}, \tau, C)}$:
	$\mathbf{k}_\tau := \sum_{j=1}^{\lambda} \mathbf{k}_{j,\tau_j}$
	Return $K := C \cdot \mathbf{k}_\tau$

Fig. 4. $\mathsf{KEM}_{\mathsf{PCA}}$, an IND-PCA-secure KEM under the $\mathcal{U}_k$-MDDH Assumption, with tag-space $\mathcal{T} = \{0,1\}^\lambda$. Here, GGen is a prime-order group generator (see Section 2.3).

game	$\mathbf{y}$ uniform in:	$\mathbf{k}'_\tau$ used by EncO and DecO	justification/remark
G_0	$\mathsf{span}(\mathbf{M})$	$\mathbf{k}_\tau$	actual scheme
G_1	$\boxed{\mathbb{Z}_q^{3k}}$	$\mathbf{k}_\tau$	$\mathcal{U}_{3k,k}$-MDDH on $[\mathbf{M}]$
$\mathrm{G}_{2.i}$	$\mathbb{Z}_q^{3k}$	$\mathbf{k}_\tau + \boxed{\mathbf{M}^\perp \mathsf{RF}_i(\tau_{\mid i})}$	$\mathrm{G}_1 \equiv \mathrm{G}_{2.0}$
$\mathrm{G}_{2.i.1}$ $\tau_{i+1} = 0$:	$\boxed{\mathsf{span}(\mathbf{M}, \mathbf{M}_0)}$	$\mathbf{k}_\tau + \mathbf{M}^\perp \mathsf{RF}_i(\tau_{\mid i})$	$\mathcal{U}_{3k,k}$-MDDH on $[\mathbf{M}_0]$
$\tau_{i+1} = 1$:	$\boxed{\mathsf{span}(\mathbf{M}, \mathbf{M}_1)}$		$\mathcal{U}_{3k,k}$-MDDH on $[\mathbf{M}_1]$
$\mathrm{G}_{2.i.2}$ $\tau_{i+1} = 0$:	$\mathsf{span}(\mathbf{M}, \mathbf{M}_0)$	$\mathbf{k}_\tau + \boxed{\mathbf{M}_0^* \mathsf{RF}_{i+1}^{(0)}(\tau_{\mid i+1})} + \mathbf{M}_1^* \mathsf{RF}_i^{(1)}(\tau_{\mid i})$	Cramer-Shoup argument
$\tau_{i+1} = 1$:	$\mathsf{span}(\mathbf{M}, \mathbf{M}_1)$		
$\mathrm{G}_{2.i.3}$ $\tau_{i+1} = 0$:	$\mathsf{span}(\mathbf{M}, \mathbf{M}_0)$	$\mathbf{k}_\tau + \mathbf{M}_0^* \mathsf{RF}_{i+1}^{(0)}(\tau_{\mid i+1}) + \boxed{\mathbf{M}_1^* \mathsf{RF}_{i+1}^{(1)}(\tau_{\mid i+1})}$	Cramer-Shoup argument
$\tau_{i+1} = 1$:	$\mathsf{span}(\mathbf{M}, \mathbf{M}_1)$		
$\mathrm{G}_{2.i+1}$	$\boxed{\mathbb{Z}_q^{3k}}$	$\mathbf{k}_\tau + \mathbf{M}^\perp \mathsf{RF}_{i+1}(\tau_{\mid i+1})$	$\mathcal{U}_{3k,k}$-MDDH on $[\mathbf{M}_0]$ and $[\mathbf{M}_1]$

Fig. 5. Sequence of games for the proof of Theorem 1. Throughout, we have (i) $\mathbf{k}_\tau := \sum_{j=1}^{\lambda} \mathbf{k}_{j,\tau_j}$; (ii) $\mathsf{EncO}(\tau) = ([\mathbf{y}], K_b)$ where $K_0 = [\mathbf{y}^\top \mathbf{k}'_\tau]$ and $K_1 \leftarrow_{\mathrm{R}} \mathbb{G}$; (iii) $\mathsf{DecO}(\tau, [\mathbf{y}], \widehat{K})$ computes the encapsulation key $K := [\mathbf{y}^\top \cdot \mathbf{k}'_\tau]$. Here, $(\mathbf{M}_0^*, \mathbf{M}_1^*)$ is a basis for $\mathsf{span}(\mathbf{M}^\perp)$, so that $\mathbf{M}_1^\perp \mathbf{M}_0^* = \mathbf{M}_0^\perp \mathbf{M}_1^* = \mathbf{0}$, and we write $\mathbf{M}^\perp \mathsf{RF}_i(\tau_{\mid i}) := \mathbf{M}_0^* \mathsf{RF}_i^{(0)}(\tau_{\mid i}) + \mathbf{M}_0^* \mathsf{RF}_i^{(1)}(\tau_{\mid i})$. The second column shows which set $\mathbf{y}$ is uniformly picked from by EncO, the third column shows the value of $\mathbf{k}'_\tau$ used by both EncO and DecO.

Namely, for any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A}) + (Q_{\mathsf{dec}} + Q_{\mathsf{enc}}) \cdot \mathsf{poly}(\lambda)$ *and*

$$\mathbf{Adv}^{\mathsf{ind\text{-}pca}}_{\mathsf{KEM}_{\mathsf{PCA}}}(\mathcal{A}) \le (4\lambda + 1) \cdot \mathbf{Adv}^{\mathrm{mddh}}_{\mathcal{U}_k, \mathsf{GGen}}(\mathcal{B}) + (Q_{\mathsf{dec}} + Q_{\mathsf{enc}}) \cdot 2^{-\Omega(\lambda)},$$

where Q_{enc}, Q_{dec} are the number of times $\mathcal{A}$ queries EncO, DecO, respectively, and $\mathsf{poly}(\lambda)$ is independent of $\mathbf{T}(\mathcal{A})$.

Proof of Theorem 1. Perfect correctness follows readily from the fact that for all $\mathbf{r} \in \mathbb{Z}_q^k$ and $C = \mathbf{r}^\top \mathbf{M}^\top$, for all $\mathbf{k} \in \mathbb{Z}_q^{3k}$:

$$\mathbf{r}^\top (\mathbf{M}^\top \mathbf{k}) = C \cdot \mathbf{k}.$$

We now prove the IND-PCA security of $\mathsf{KEM}_{\mathsf{PCA}}$. We proceed via a series of games described in Figure 6 and 7 and we use $\mathbf{Adv}_i$ to denote the advantage of $\mathcal{A}$ in game G_i. We also give a high-level picture of the proof in Figure 5, summarizing the sequence of games.

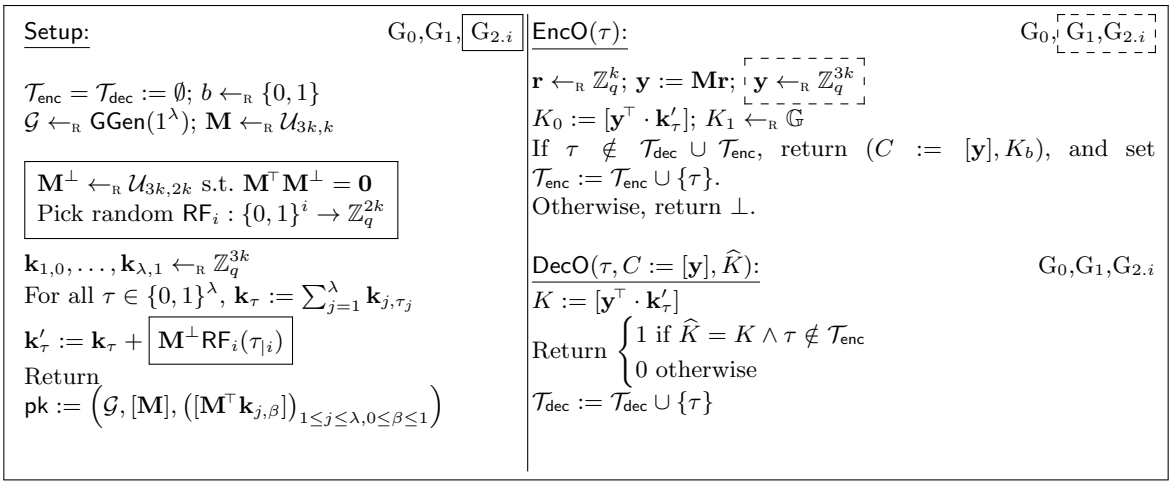

Fig. 6. Games $\mathrm{G}_0, \mathrm{G}_1, \mathrm{G}_{2.i}$ (for $0 \le i \le \lambda$) for the proof of multi-ciphertext PCA security of $\mathsf{KEM}_{\mathsf{PCA}}$ in Figure 4. In each procedure, the components inside a solid (dotted) frame are only present in the games marked by a solid (dotted) frame.

Lemma 4 (G_0 to G_1). *There exists an adversary $\mathcal{B}_0$ such that $\mathbf{T}(\mathcal{B}_0) \approx \mathbf{T}(\mathcal{A}) + (Q_{\mathsf{enc}} + Q_{\mathsf{dec}}) \cdot \mathsf{poly}(\lambda)$ and*

$$|\mathbf{Adv}_0 - \mathbf{Adv}_1| \le \mathbf{Adv}^{\mathrm{mddh}}_{\mathcal{U}_k, \mathsf{GGen}}(\mathcal{B}_0) + \frac{1}{q-1},$$

where Q_{enc}, Q_{dec} are the number of times $\mathcal{A}$ queries EncO, DecO, respectively, and $\mathsf{poly}(\lambda)$ is independent of $\mathbf{T}(\mathcal{A})$.

Here, we use the MDDH assumption to "tightly" switch the distribution of all the challenge ciphertexts.

Proof of Lemma 4. To go from G_0 to G_1, we switch the distribution of the vectors $[\mathbf{y}]$ sampled by EncO, using the Q_{enc}-fold $\mathcal{U}_{3k,k}$-MDDH Assumption on $[\mathbf{M}]$ (see Definition 4 and Lemma 3).

We build an adversary $\mathcal{B}'_0$ against the Q_{enc}-fold $\mathcal{U}_{3k,k}$-MDDH Assumption, such that $\mathbf{T}(\mathcal{B}'_0) \approx \mathbf{T}(\mathcal{A}) + (Q_{\mathsf{enc}} + Q_{\mathsf{dec}}) \cdot \mathsf{poly}(\lambda)$ with $\mathsf{poly}(\lambda)$ independent of $\mathbf{T}(\mathcal{A})$, and

$$|\mathbf{Adv}_0 - \mathbf{Adv}_1| \le \mathbf{Adv}^{\mathrm{Q_{enc}\text{-}mddh}}_{\mathcal{U}_{3k,k}, \mathsf{GGen}}(\mathcal{B}'_0).$$

This implies the lemma by Lemma 3 (self-reducibility of $\mathcal{U}_{3k,k}$-MDDH), and Lemma 1 ($\mathcal{U}_{3k,k}$-$\mathsf{MDDH} \Leftrightarrow \mathcal{U}_k$-$\mathsf{MDDH}$).

Upon receiving a challenge $(\mathcal{G}, [\mathbf{M}] \in \mathbb{G}^{3k \times k}, [\mathbf{H}] := [\mathbf{h}_1 | \ldots | \mathbf{h}_{Q_{\mathsf{enc}}}] \in \mathbb{G}^{3k \times Q_{\mathsf{enc}}})$ for the Q_{enc}-fold $\mathcal{U}_{3k,k}$-MDDH Assumption, $\mathcal{B}'_0$ picks $b \leftarrow_{\mathrm{R}} \{0,1\}$, $\mathbf{k}_{1,0}, \ldots, \mathbf{k}_{\lambda,1} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k}$, and simulates Setup, DecO as described in Figure 6. To simulate EncO on its j'th query, for $j = 1, \ldots, Q_{\mathsf{enc}}$, $\mathcal{B}'_0$ sets $[\mathbf{y}] := [\mathbf{h}_j]$, and computes K_b as described in Figure 6. □

Lemma 5 (G_1 to $\mathrm{G}_{2.0}$). $|\mathbf{Adv}_1 - \mathbf{Adv}_{2.0}| = 0$.

Proof of Lemma 5. We show that the two games are statistically equivalent. To go from G_1 to $\mathrm{G}_{2.0}$, we change the distribution of $\mathbf{k}_{1,\beta} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k}$ for $\beta = 0, 1$, to $\mathbf{k}_{1,\beta} + \mathbf{M}^\perp \mathsf{RF}_0(\varepsilon)$, where $\mathbf{k}_{1,\beta} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k}$,

$\mathsf{RF}_0(\varepsilon) \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{2k}$, and $\mathbf{M}^{\perp} \leftarrow_{\mathrm{R}} \mathcal{U}_{3k,2k}$ such that $\mathbf{M}^{\top}\mathbf{M}^{\perp} = \mathbf{0}$. Note that the extra term $\mathbf{M}^{\perp}\mathsf{RF}_0(\varepsilon)$ does not appear in pk, since $\mathbf{M}^{\top}(\mathbf{k}_{1,\beta} + \mathbf{M}^{\perp}\mathsf{RF}_0(\varepsilon)) = \mathbf{M}^{\top}\mathbf{k}_{1,\beta}$. □

Lemma 6 ($\mathrm{G}_{2.i}$ to $\mathrm{G}_{2.i+1}$). *For all $0 \le i \le \lambda - 1$, there exists an adversary $\mathcal{B}_{2.i}$ such that $\mathbf{T}(\mathcal{B}_{2.i}) \approx \mathbf{T}(\mathcal{A}) + (Q_{\mathsf{enc}} + Q_{\mathsf{dec}}) \cdot \mathsf{poly}(\lambda)$ and*

$$|\mathbf{Adv}_{2.i} - \mathbf{Adv}_{2.i+1}| \le 4 \cdot \mathbf{Adv}^{\mathrm{mddh}}_{\mathcal{U}_k,\mathsf{GGen}}(\mathcal{B}_{2.i}) + \frac{4Q_{\mathsf{dec}} + 2k}{q} + \frac{4}{q-1},$$

where Q_{enc}, Q_{dec} are the number of times $\mathcal{A}$ queries EncO, DecO, respectively, and $\mathsf{poly}(\lambda)$ is independent of $\mathbf{T}(\mathcal{A})$.

Proof of Lemma 6. To go from $\mathrm{G}_{2.i}$ to $\mathrm{G}_{2.i+1}$, we introduce intermediate games $\mathrm{G}_{2.i.1}$, $\mathrm{G}_{2.i.2}$ and $\mathrm{G}_{2.i.3}$, defined in Figure 7. We prove that these games are indistinguishable in Lemma 7, 8, 9, and 10.

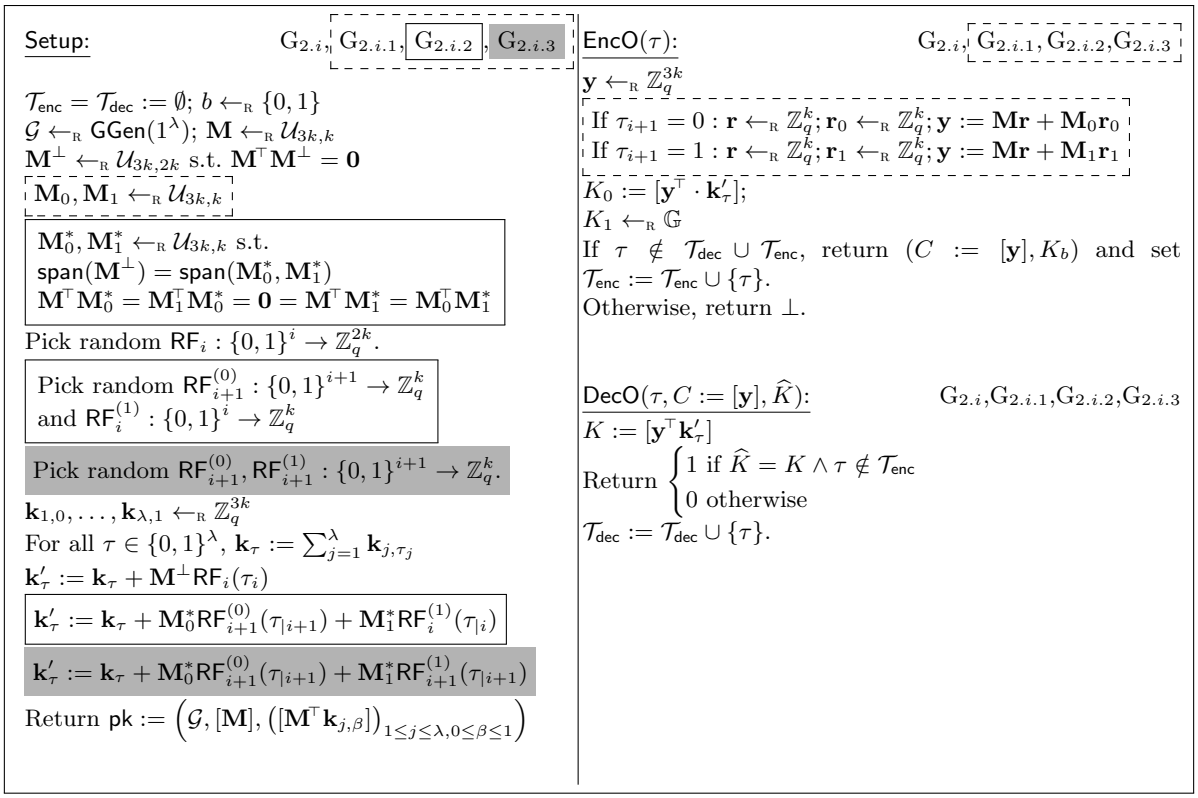

Fig. 7. Games $\mathrm{G}_{2.i}$ (for $0 \le i \le \lambda$),$\mathrm{G}_{2.i.1}$, $\mathrm{G}_{2.i.2}$ and $\mathrm{G}_{2.i.3}$ (for $0 \le i \le \lambda - 1$) for the proof of Lemma 6. For all $\tau \in \{0,1\}^{\lambda}$, we denote by $\tau_{|i}$ the i-bit prefix of τ. In each procedure, the components inside a solid (dotted, gray) frame are only present in the games marked by a solid (dotted, gray) frame.

Lemma 7 ($\mathrm{G}_{2.i}$ to $\mathrm{G}_{2.i.1}$). *For all $0 \le i \le \lambda - 1$, there exists an adversary $\mathcal{B}_{2.i.0}$ such that $\mathbf{T}(\mathcal{B}_{2.i.0}) \approx \mathbf{T}(\mathcal{A}) + (Q_{\mathsf{enc}} + Q_{\mathsf{dec}}) \cdot \mathsf{poly}(\lambda)$ and*

$$|\mathbf{Adv}_{2.i} - \mathbf{Adv}_{2.i.1}| \le 2 \cdot \mathbf{Adv}^{\mathrm{mddh}}_{\mathcal{U}_k,\mathsf{GGen}}(\mathcal{B}_{2.i.0}) + \frac{2}{q-1},$$

where Q_{enc}, Q_{dec} are the number of times $\mathcal{A}$ queries EncO, DecO, *respectively, and* $\mathsf{poly}(\lambda)$ *is independent of* $\mathbf{T}(\mathcal{A})$.

Here, we use the MDDH Assumption to "tightly" switch the distribution of all the challenge ciphertexts. We proceed in two steps, first, by changing the distribution of all the ciphertexts with a tag τ such that $\tau_{i+1} = 0$, and then, for those with a tag τ such that $\tau_{i+1} = 1$. We use the MDDH Assumption with respect to an independent matrix for each step.

Proof of Lemma 7. To go from $\mathrm{G}_{2.i}$ to $\mathrm{G}_{2.i.1}$, we switch the distribution of the vectors $[\mathbf{y}]$ sampled by EncO, using the Q_{enc}-fold $\mathcal{U}_{3k,k}$-MDDH Assumption.

We introduce an intermediate game $\mathrm{G}_{2.i.0}$ where $\mathsf{EncO}(\tau)$ is computed as in $\mathrm{G}_{2.i.1}$ if $\tau_{i+1} = 0$, and as in $\mathrm{G}_{2.i}$ if $\tau_{i+1} = 1$. Setup, DecO are as in $\mathrm{G}_{2.i.1}$. We build adversaries $\mathcal{B}'_{2.i.0}$ and $\mathcal{B}''_{2.i.0}$ such that $\mathbf{T}(\mathcal{B}'_{2.i.0}) \approx \mathbf{T}(\mathcal{B}''_{2.i.0}) \approx \mathbf{T}(\mathcal{A}) + (Q_{\mathsf{enc}} + Q_{\mathsf{dec}}) \cdot \mathsf{poly}(\lambda)$ with $\mathsf{poly}(\lambda)$ independent of $\mathbf{T}(\mathcal{A})$, and

Claim 1: $|\mathbf{Adv}_{2.i} - \mathbf{Adv}_{2.i.0}| \leq \mathbf{Adv}^{\mathrm{Q_{enc}\text{-}mddh}}_{\mathcal{U}_{3,k},\mathsf{GGen}}(\mathcal{B}'_{2.i.0})$.

Claim 2: $|\mathbf{Adv}_{2.i.0} - \mathbf{Adv}_{2.i.1}| \leq \mathbf{Adv}^{\mathrm{Q_{enc}\text{-}mddh}}_{\mathcal{U}_{3k,k},\mathsf{GGen}}(\mathcal{B}''_{2.i.0})$.

This implies the lemma by Lemma 3 (self-reducibility of $\mathcal{U}_{3k,k}$-MDDH), and Lemma 1 ($\mathcal{U}_{3k,k}$-MDDH $\Leftrightarrow \mathcal{U}_k$-MDDH).

Let us prove Claim 1. Upon receiving a challenge $(\mathcal{G}, [\mathbf{M}_0] \in \mathbb{G}^{3k\times k}, [\mathbf{H}] := [\mathbf{h}_1|\dots|\mathbf{h}_{Q_{\mathsf{enc}}}] \in \mathbb{G}^{3k\times Q_{\mathsf{enc}}})$ for the Q_{enc}-fold $\mathcal{U}_{3k,k}$-MDDH Assumption with respect to $\mathbf{M}_0 \leftarrow_{\mathrm{R}} \mathcal{U}_{3k,k}$, $\mathcal{B}'_{2.i.0}$ does as follows:

Setup**:** $\mathcal{B}'_{2.i.0}$ picks $\mathbf{M} \leftarrow_{\mathrm{R}} \mathcal{U}_{3k,k}$, $\mathbf{k}_{1,0}, \dots, \mathbf{k}_{\lambda,1} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k}$, and computes pk as described in Figure 7. For each τ queried to EncO or DecO, it computes on the fly $\mathsf{RF}_i(\tau_{|i})$ and $\mathbf{k}'_\tau := \mathbf{k}_\tau + \mathbf{M}^\perp \mathsf{RF}_i(\tau_{|i})$, where $\mathbf{k}_\tau := \sum_{j=1}^{\lambda} \mathbf{k}_{j,\tau_j}$, $\mathsf{RF}_i : \{0,1\}^i \to \mathbb{Z}_q^{2k}$ is a random function, and $\tau_{|i}$ denotes the i-bit prefix of τ (see Figure 7). Note that $\mathcal{B}'_{2.i.0}$ can compute efficiently $\mathbf{M}^\perp$ from $\mathbf{M}$.

EncO**:** To simulate the oracle $\mathsf{EncO}(\tau)$ on its j'th query, for $j = 1, \dots, Q_{\mathsf{enc}}$, $\mathcal{B}'_{2.i.0}$ computes $[\mathbf{y}]$ as follows:

$$\begin{array}{l} \text{if } \tau_{i+1} = 0 : \mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k;\ [\mathbf{y}] := [\mathbf{M}\mathbf{r} + \mathbf{h}_j] \\ \text{if } \tau_{i+1} = 1 : [\mathbf{y}] \leftarrow_{\mathrm{R}} \mathbb{G}^{3k} \end{array}$$

This way, $\mathcal{B}'_{2.i.0}$ simulates EncO as in $\mathrm{G}_{2.i.0}$ when $[\mathbf{h}_j] := [\mathbf{M}_0\mathbf{r}_0]$ with $\mathbf{r}_0 \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$, and as in $\mathrm{G}_{2.i}$ when $[\mathbf{h}_j] \leftarrow_{\mathrm{R}} \mathbb{G}^{3k}$.

DecO**:** Finally, $\mathcal{B}'_{2.i.0}$ simulates DecO as described in Figure 7.

Therefore, $|\mathbf{Adv}_{2.i} - \mathbf{Adv}_{2.i.0}| \leq \mathbf{Adv}^{\mathrm{Q_{enc}\text{-}mddh}}_{\mathcal{U}_{3k,k},\mathsf{GGen}}(\mathcal{B}'_{2.i.0})$.

To prove Claim 2, we build an adversary $\mathcal{B}''_{2.i.0}$ against the Q_{enc}-fold $\mathcal{U}_{3k,k}$-MDDH Assumption with respect to a matrix $\mathbf{M}_1 \leftarrow_{\mathrm{R}} \mathcal{U}_{3k,k}$, independent from $\mathbf{M}_0$, similarly than $\mathcal{B}'_{2.i.0}$. □

Lemma 8 ($\mathrm{G}_{2.i.1}$ **to** $\mathrm{G}_{2.i.2}$)**.** *For all* $0 \leq i \leq \lambda - 1$,

$$|\mathbf{Adv}_{2.i.1} - \mathbf{Adv}_{2.i.2}| \leq \frac{2Q_{\mathsf{dec}} + 2k}{q},$$

where Q_{dec} is the number of times $\mathcal{A}$ queries DecO.

Here, we use a variant of the Cramer-Shoup information-theoretic argument to move from RF_i to RF_{i+1}, thereby increasing the entropy of $\mathbf{k}'_\tau$ computed by Setup. For the sake of readability, we proceed in two steps: in Lemma 8, we move from RF_i to an hybrid between RF_i and RF_{i+1}, and in Lemma 9, we move to RF_{i+1}.

Proof of Lemma 8. In $\mathrm{G}_{2.i.2}$, we decompose $\mathsf{span}(\mathbf{M}^\perp)$ into two subspaces $\mathsf{span}(\mathbf{M}_0^*)$ and $\mathsf{span}(\mathbf{M}_1^*)$, and we increase the entropy of the components of $\mathbf{k}'_\tau$ which lie in $\mathsf{span}(\mathbf{M}_0^*)$. To argue that $\mathrm{G}_{2.i.1}$ and $\mathrm{G}_{2.i.2}$ are statistically close, we use a Cramer-Shoup argument [11].

Let us first explain how the matrices $\mathbf{M}_0^*$ and $\mathbf{M}_1^*$ are sampled. Note that with probability at least $1 - \frac{2k}{q}$ over the random coins of Setup, $(\mathbf{M}\|\mathbf{M}_0\|\mathbf{M}_1)$ forms a basis of $\mathbb{Z}_q^{3k}$. Therefore, we have $\mathsf{span}(\mathbf{M}^\perp) = \mathsf{Ker}(\mathbf{M}^\top) = \mathsf{Ker}\big((\mathbf{M}\|\mathbf{M}_1)^\top\big) \oplus \mathsf{Ker}\big((\mathbf{M}\|\mathbf{M}_0)^\top\big)$. We pick uniformly $\mathbf{M}_0^*$ and $\mathbf{M}_1^*$ in $\mathbb{Z}_q^{3k\times k}$ that generate $\mathsf{Ker}\big((\mathbf{M}\|\mathbf{M}_1)^\top\big)$ and $\mathsf{Ker}\big((\mathbf{M}\|\mathbf{M}_0)^\top\big)$, respectively (see Figure 1.1). This way, for all $\tau \in \{0,1\}^\lambda$, we can write

$$\mathbf{M}^\perp \mathsf{RF}_i(\tau_{|i}) := \mathbf{M}_0^* \mathsf{RF}_i^{(0)}(\tau_{|i}) + \mathbf{M}_1^* \mathsf{RF}_i^{(1)}(\tau_{|i}),$$

where $\mathsf{RF}_i^{(0)}$, $\mathsf{RF}_i^{(1)} : \{0,1\}^i \to \mathbb{Z}_q^k$ are independent random functions.

We define $\mathsf{RF}_{i+1}^{(0)} : \{0,1\}^{i+1} \to \mathbb{Z}_q^k$ as follows:

$$\mathsf{RF}_{i+1}^{(0)}(\tau_{|i+1}) := \begin{cases} \mathsf{RF}_i^{(0)}(\tau_{|i}) & \text{if } \tau_{i+1} = 0 \\ \mathsf{RF}_i^{(0)}(\tau_{|i}) + \mathsf{RF}'^{(0)}_i(\tau_{|i}) & \text{if } \tau_{i+1} = 1 \end{cases}$$

where $\mathsf{RF}'^{(0)}_i : \{0,1\}^i \to \mathbb{Z}_q^k$ is a random function independent from $\mathsf{RF}_i^{(0)}$. This way, $\mathsf{RF}_{i+1}^{(0)}$ is a random function.

We show that the outputs of EncO and DecO are statistically close in $\mathrm{G}_{2.i.1}$ and $\mathrm{G}_{2.i.2}$. We decompose the proof in two cases (delimited with ■): the queries with a tag $\tau \in \{0,1\}^\lambda$ such that $\tau_{i+1} = 0$, and the queries with a tag τ such that $\tau_{i+1} = 1$.

Queries with $\tau_{i+1} = 0$:

The only difference between $\mathrm{G}_{2.i.1}$ and $\mathrm{G}_{2.i.2}$ is that Setup computes $\mathbf{k}'_\tau$ using the random function $\mathsf{RF}_i^{(0)}$ in $\mathrm{G}_{2.i.1}$, whereas it uses the random function $\mathsf{RF}_{i+1}^{(0)}$ in $\mathrm{G}_{2.i.2}$ (see Figure 7). Therefore, by definition of $\mathsf{RF}_{i+1}^{(0)}$, for all $\tau \in \{0,1\}^\lambda$ such that $\tau_{i+1} = 0$, $\mathbf{k}'_\tau$ is the same in $\mathrm{G}_{2.i.1}$ and $\mathrm{G}_{2.i.2}$, and the outputs of EncO and DecO are identically distributed. ■

Queries with $\tau_{i+1} = 1$:

Observe that for all $\mathbf{y} \in \mathsf{span}(\mathbf{M}, \mathbf{M}_1)$ and all $\tau \in \{0,1\}^\lambda$ such that $\tau_{i+1} = 1$,

$$\begin{aligned} &\overbrace{\mathbf{y}^\top \Big(\mathbf{k}_\tau + \mathbf{M}_0^* \mathsf{RF}_i^{(0)}(\tau_{|i}) + \mathbf{M}_1^* \mathsf{RF}_i^{(1)}(\tau_{|i}) + \boxed{\mathbf{M}_0^* \mathsf{RF}'^{(0)}_i(\tau_{|i})}\Big)}^{\mathrm{G}_{2.i.2}} \\ &= \mathbf{y}^\top \Big(\mathbf{k}_\tau + \mathbf{M}_0^* \mathsf{RF}_i^{(0)}(\tau_{|i}) + \mathbf{M}_1^* \mathsf{RF}_i^{(1)}(\tau_{|i})\Big) + \underbrace{\boxed{\mathbf{y}^\top \mathbf{M}_0^* \mathsf{RF}'^{(0)}_i(\tau_{|i})}}_{=0} \\ &= \overbrace{\mathbf{y}^\top \cdot \Big(\mathbf{k}_\tau + \mathbf{M}_0^* \mathsf{RF}_i^{(0)}(\tau_{|i}) + \mathbf{M}_1^* \mathsf{RF}_i^{(1)}(\tau_{|i})\Big)}^{\mathrm{G}_{2.i.1}} \end{aligned}$$

where the second equality uses the fact that $\mathbf{M}^\top \mathbf{M}_0^* = \mathbf{M}_1^\top \mathbf{M}_0^* = \mathbf{0}$ and thus $\mathbf{y}^\top \mathbf{M}_0^* = \mathbf{0}$.

This means that:

– the output of EncO on any input τ such that $\tau_{i+1} = 1$ is identically distributed in $\mathrm{G}_{2.i.1}$ and $\mathrm{G}_{2.i.2}$;
– the output of DecO on any input $(\tau, [\mathbf{y}], \widehat{K})$ where $\tau_{i+1} = 1$, and $\mathbf{y} \in \mathsf{span}(\mathbf{M}, \mathbf{M}_1)$ is the same in $\mathrm{G}_{2.i.1}$ and $\mathrm{G}_{2.i.2}$.

Henceforth, we focus on the *ill-formed* queries to DecO, namely those corresponding to $\tau_{i+1} = 1$, and $\mathbf{y} \notin \mathsf{span}(\mathbf{M}, \mathbf{M}_1)$. We introduce intermediate games $\mathrm{G}_{2.i.1.j}$, and $\mathrm{G}'_{2.i.1.j}$ for $j = 0, \ldots, Q_{\mathsf{dec}}$, defined as follows:

– $\mathrm{G}_{2.i.1.j}$: DecO is as in $\mathrm{G}_{2.i.1}$ except that for the first j times it is queried, it outputs 0 to any ill-formed query. EncO is as in $\mathrm{G}_{2.i.2}$.
– $\mathrm{G}'_{2.i.1.j}$: DecO as in $\mathrm{G}_{2.i.2}$ except that for the first j times it is queried, it outputs 0 to any ill-formed query. EncO is as in $\mathrm{G}_{2.i.2}$.

We show that:

$$\mathrm{G}_{2.i.1} \equiv \mathrm{G}_{2.i.1.0} \approx_s \mathrm{G}_{2.i.1.1} \approx_s \ldots \approx_s \mathrm{G}_{2.i.1.Q_{\mathsf{dec}}} \equiv \mathrm{G}'_{2.i.1.Q_{\mathsf{dec}}} \approx_s \mathrm{G}'_{2.i.1.Q_{\mathsf{dec}}-1} \approx_s \ldots \approx_s \mathrm{G}'_{2.i.1.0} \equiv \mathrm{G}_{2.i.2}$$

where we denote statistical closeness with $\approx_s$ and statistical equality with $\equiv$.

It suffices to show that for all $j = 0, \ldots, Q_{\mathsf{dec}} - 1$:

Claim 1: in $\mathrm{G}_{2.i.1.j}$, if the $j+1$-st query is ill-formed, then DecO outputs 0 with overwhelming probability $1 - 1/q$ (this implies $\mathrm{G}_{2.i.1.j} \approx_s \mathrm{G}_{2.i.1.j+1}$, with statistical difference $1/q$);
Claim 2: in $\mathrm{G}'_{2.i.1.j}$, if the $j+1$-st query is ill-formed, then DecO outputs 0 with overwhelming probability $1 - 1/q$ (this implies $\mathrm{G}'_{2.i.1.j} \approx_s \mathrm{G}'_{2.i.1.j+1}$, with statistical difference $1/q$)

where the probabilities are taken over the random coins of Setup.

Let us prove Claim 1. Recall that in $\mathrm{G}_{2.i.1.j}$, on its $j+1$-st query, $\mathsf{DecO}(\tau, [\mathbf{y}], \widehat{K})$ computes $K := [\mathbf{y}^\top \mathbf{k}'_\tau]$, where $\mathbf{k}'_\tau := \left(\mathbf{k}_\tau + \mathbf{M}_0^* \mathsf{RF}_i^{(0)}(\tau_{|i}) + \mathbf{M}_1^* \mathsf{RF}_i^{(1)}(\tau_{|i})\right)$ (see Figure 7). We prove that if $(\tau, [\mathbf{y}], \widehat{K})$ is ill-formed, then K is completely hidden from $\mathcal{A}$, up to its $j+1$-st query to DecO. The reason is that the vector $\mathbf{k}_{i+1,1}$ in sk contains some entropy that is hidden from $\mathcal{A}$. This entropy is "released" on the $j+1$-st query to DecO if it is ill-formed. More formally, we use the fact that the vector $\mathbf{k}_{i+1,1} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k}$ is identically distributed as $\mathbf{k}_{i+1,1} + \mathbf{M}_0^* \mathbf{w}$, where $\mathbf{k}_{i+1,1} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k}$, and $\mathbf{w} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$. We show that $\mathbf{w}$ is completely hidden from $\mathcal{A}$, up to its $j+1$-st query to DecO.

– The public key pk does not leak any information about $\mathbf{w}$, since

$$\mathbf{M}^\top (\mathbf{k}_{i+1,1} + \boxed{\mathbf{M}_0^* \mathbf{w}}) = \mathbf{M}^\top \mathbf{k}_{i+1,1}.$$

This is because $\mathbf{M}^\top \mathbf{M}_0^* = \mathbf{0}$.
– The outputs of EncO also hide $\mathbf{w}$.
  • For τ such that $\tau_{i+1} = 0$, $\mathbf{k}'_\tau$ is independent of $\mathbf{k}_{i+1,1}$, and therefore, so does $\mathsf{EncO}(\tau)$.
  • For τ such that $\tau_{i+1} = 1$, and for any $\mathbf{y} \in \mathsf{span}(\mathbf{M}, \mathbf{M}_1)$, we have:

$$\mathbf{y}^\top (\mathbf{k}'_\tau + \boxed{\mathbf{M}_0^* \mathbf{w}}) = \mathbf{y}^\top \mathbf{k}'_\tau \tag{2}$$

since $\mathbf{M}^\top \mathbf{M}_0^* = \mathbf{M}_1^\top \mathbf{M}_0^* = \mathbf{0}$, which implies $\mathbf{y}^\top \mathbf{M}_0^* = \mathbf{0}$.
– The first j outputs of DecO also hide $\mathbf{w}$.
  • For τ such that $\tau_{i+1} = 0$, $\mathbf{k}'_\tau$ is independent of $\mathbf{k}_{i+1,1}$, and therefore, so does $\mathsf{DecO}([\mathbf{y}], \tau, \widehat{K})$.

- For τ such that $\tau_{i+1} = 1$ and $\mathbf{y} \in \mathsf{span}(\mathbf{M}, \mathbf{M}_1)$, the fact that $\mathsf{DecO}(\tau, [\mathbf{y}], \widehat{K})$ is independent of $\mathbf{w}$ follows readily from Equation (2).
- For τ such that $\tau_{i+1} = 1$ and $\mathbf{y} \notin \mathsf{span}(\mathbf{M}, \mathbf{M}_1)$, that is, for an ill-formed query, DecO outputs 0, independently of $\mathbf{w}$, by definition of $\mathrm{G}_{2.i.1.j}$.

This proves that $\mathbf{w}$ is uniformly random from $\mathcal{A}$'s viewpoint.

Finally, because the j+1-st query $(\tau, [\mathbf{y}], \widehat{K})$ is ill-formed, we have $\tau_{i+1} = 1$, and $\mathbf{y} \notin \mathsf{span}(\mathbf{M}, \mathbf{M}_1)$, which implies that $\mathbf{y}^\top \mathbf{M}_0^* \neq \mathbf{0}$. Therefore, the value

$$K = [\mathbf{y}^\top(\mathbf{k}'_\tau + \mathbf{M}_0^* \mathbf{w})] = [\mathbf{y}^\top \mathbf{k}'_\tau + \underbrace{\mathbf{y}^\top \mathbf{M}_0^*}_{\neq \mathbf{0}} \mathbf{w}]$$

computed by DecO is uniformly random over $\mathbb{G}$ from $\mathcal{A}$'s viewpoint. Thus, with probability $1 - 1/q$ over $K \leftarrow_{\mathrm{R}} \mathbb{G}$, we have $\widehat{K} \neq K$, and $\mathsf{DecO}(\tau, [\mathbf{y}], \widehat{K}) = 0$.

We prove Claim 2 similarly, arguing than in $\mathrm{G}'_{2.i.1.j}$, the value $K := [\mathbf{y}^\top \mathbf{k}'_\tau]$, where $\mathbf{k}'_\tau := \big(\mathbf{k}_\tau + \mathbf{M}_0^* \mathsf{RF}_{i+1}^{(0)}(\tau_{|i+1}) + \mathbf{M}_1^* \mathsf{RF}_i^{(1)}(\tau_{|i})\big)$, computed by $\mathsf{DecO}(\tau, [\mathbf{y}], \widehat{K})$ on its $j+1$-st query, is completely hidden from $\mathcal{A}$, up to its $j+1$-st query to DecO, if $(\tau, [\mathbf{y}], \widehat{K})$ is ill-formed. The argument goes exactly as for Claim 1. ■ □

Lemma 9 ($\mathrm{G}_{2.i.2}$ **to** $\mathrm{G}_{2.i.3}$)**.** *For all* $0 \leq i \leq \lambda - 1$*,*

$$|\mathbf{Adv}_{2.i.2} - \mathbf{Adv}_{2.i.3}| \leq \frac{2Q_{\mathsf{dec}}}{q},$$

where Q_{dec} *is the number of times* $\mathcal{A}$ *queries* DecO*.*

Proof of Lemma 9. In $\mathrm{G}_{2.i.3}$, we use the same decomposition $\mathsf{span}(\mathbf{M}^\perp) = \mathsf{span}(\mathbf{M}_0^*, \mathbf{M}_1^*)$ as that in $\mathrm{G}_{2.i.2}$. The entropy of the components of $\mathbf{k}'_\tau$ that lie in $\mathsf{span}(\mathbf{M}_1^*)$ increases from $\mathrm{G}_{2.i.2}$ to $\mathrm{G}_{2.i.3}$. To argue that these two games are statistically close, we use a Cramer-Shoup argument [11], exactly as for Lemma 8.

We define $\mathsf{RF}_{i+1}^{(1)}\{0,1\}^{i+1} \to \mathbb{Z}_q^k$ as follows:

$$\mathsf{RF}_{i+1}^{(1)}(\tau_{|i+1}) := \begin{cases} \mathsf{RF}_i^{(1)}(\tau_{|i}) + \mathsf{RF}'^{(1)}_i(\tau_{|i}) & \text{if } \tau_{i+1} = 0 \\ \mathsf{RF}_i^{(1)}(\tau_{|i}) & \text{if } \tau_{i+1} = 1 \end{cases}$$

where $\mathsf{RF}'^{(1)}_i : \{0,1\}^i \to \mathbb{Z}_q^k$ is a random function independent from $\mathsf{RF}_i^{(1)}$. This way, $\mathsf{RF}_{i+1}^{(1)}$ is a random function.

We show that the outputs of EncO and DecO are statistically close in $\mathrm{G}_{2.i.1}$ and $\mathrm{G}_{2.i.2}$. We decompose the proof in two cases (delimited with ■): the queries with a tag $\tau \in \{0,1\}^\lambda$ such that $\tau_{i+1} = 0$, and the queries with tag τ such that $\tau_{i+1} = 1$.

Queries with $\tau_{i+1} = 1$**:**

The only difference between $\mathrm{G}_{2.i.2}$ and $\mathrm{G}_{2.i.3}$ is that Setup computes $\mathbf{k}'_\tau$ using the random function $\mathsf{RF}_i^{(1)}$ in $\mathrm{G}_{2.i.2}$, whereas it uses the random function $\mathsf{RF}_{i+1}^{(1)}$ in $\mathrm{G}_{2.i.3}$ (see Figure 7). Therefore, by definition of $\mathsf{RF}_{i+1}^{(1)}$, for all $\tau \in \{0,1\}^\lambda$ such that $\tau_{i+1} = 1$, $\mathbf{k}'_\tau$ is the same in $\mathrm{G}_{2.i.2}$ and $\mathrm{G}_{2.i.3}$, and the outputs of EncO and DecO are identically distributed. ■

Queries with $\tau_{i+1} = 0$:
Observe that for all $\mathbf{y} \in \mathsf{span}(\mathbf{M}, \mathbf{M}_0)$ and all $\tau \in \{0,1\}^\lambda$ such that $\tau_{i+1} = 0$,

$$\begin{aligned}
&\overbrace{\mathbf{y}^\top \Big(\mathbf{k}_\tau + \mathbf{M}_0^* \mathsf{RF}_{i+1}^{(0)}(\tau_{|i+1}) + \mathbf{M}_1^* \mathsf{RF}_i^{(1)}(\tau_{|i}) + \boxed{\mathbf{M}_1^* \mathsf{RF}'^{(1)}_i(\tau_{|i})}\Big)}^{\mathrm{G}_{2.i.3}} \\
&= \mathbf{y}^\top \Big(\mathbf{k}_\tau + \mathbf{M}_0^* \mathsf{RF}_{i+1}^{(0)}(\tau_{|i+1}) + \mathbf{M}_1^* \mathsf{RF}_i^{(1)}(\tau_{|i})\Big) + \underbrace{\boxed{\mathbf{y}^\top \mathbf{M}_1^* \mathsf{RF}'^{(1)}_i(\tau_{|i})}}_{=0} \\
&= \overbrace{\mathbf{y}^\top \cdot \Big(\mathbf{k}_\tau + \mathbf{M}_0^* \mathsf{RF}_{i+1}^{(0)}(\tau_{|i+1}) + \mathbf{M}_1^* \mathsf{RF}_i^{(1)}(\tau_{|i})\Big)}^{\mathrm{G}_{2.i.2}}
\end{aligned}$$

where the second equality uses the fact $\mathbf{M}^\top \mathbf{M}_1^* = \mathbf{M}_0^\top \mathbf{M}_1^* = \mathbf{0}$, which implies $\mathbf{y}^\top \mathbf{M}_1^* = \mathbf{0}$.

This means that:

- the output of EncO on any input τ such that $\tau_{i+1} = 0$ is identically distributed in $\mathrm{G}_{2.i.2}$ and $\mathrm{G}_{2.i.3}$;
- the output of DecO on any input $(\tau, [\mathbf{y}], \widehat{K})$ where $\tau_{i+1} = 0$, and $\mathbf{y} \in \mathsf{span}(\mathbf{M}, \mathbf{M}_0)$ is the same in $\mathrm{G}_{2.i.2}$ and $\mathrm{G}_{2.i.3}$.

Henceforth, we focus on the *ill-formed* queries to DecO, namely those corresponding to $\tau_{i+1} = 0$, and $\mathbf{y} \notin \mathsf{span}(\mathbf{M}, \mathbf{M}_0)$. The rest of the proof goes similarly than the proof of Lemma 8. See the latter for further details. ■ □

Lemma 10 ($\mathrm{G}_{2.i.3}$ to $\mathrm{G}_{2.i+1}$). *For all $0 \leq i \leq \lambda - 1$, there exists an adversary $\mathcal{B}_{2.i.3}$ such that $\mathbf{T}(\mathcal{B}_{2.i.3}) \approx \mathbf{T}(\mathcal{A}) + (Q_{\mathsf{enc}} + Q_{\mathsf{dec}}) \cdot \mathsf{poly}(\lambda)$ and*

$$|\mathbf{Adv}_{2.i.3} - \mathbf{Adv}_{2.i+1}| \leq 2 \cdot \mathbf{Adv}^{\mathrm{mddh}}_{\mathcal{U}_k, \mathsf{GGen}}(\mathcal{B}_{2.i.3}) + \frac{2}{q-1}$$

where Q_{enc}, Q_{dec} are the number of times $\mathcal{A}$ queries EncO, DecO, respectively, and $\mathsf{poly}(\lambda)$ is independent of $\mathbf{T}(\mathcal{A})$.

Here, we use the MDDH Assumption to "tightly" switch the distribution of all the challenge ciphertexts, as for Lemma 7. We proceed in two steps, first, by changing the distribution of all the ciphertexts with a tag τ such that $\tau_{i+1} = 0$, and then, the distribution of those with a tag τ such that $\tau_{i+1} = 1$, using the MDDH Assumption with respect to an independent matrix for each step.
Proof of Lemma 10. To go from $\mathrm{G}_{2.i.3}$ to $\mathrm{G}_{2.i+1}$, we switch the distribution of the vectors $[\mathbf{y}]$ sampled by EncO, using the Q_{enc}-fold $\mathcal{U}_{3k,k}$-MDDH Assumption. This transition is symmetric to the transition between $\mathrm{G}_{2.i}$ and $\mathrm{G}_{2.i.1}$ (see the proof of Lemma 7 for further details). Finally, we use the fact that for all $\tau \in \{0,1\}^\lambda$, $\mathbf{M}_0^* \mathsf{RF}_{i+1}^{(0)}(\tau_{|i}) + \mathbf{M}_1^* \mathsf{RF}_{i+1}^{(1)}(\tau_{|i+1})$ is identically distributed to $\mathbf{M}^\perp \mathsf{RF}_{i+1}(\tau_{|i+1})$, where $\mathsf{RF}_{i+1} : \{0,1\}^{i+1} \to \mathbb{Z}_q^{2k}$ is a random function. This is because $(\mathbf{M}_0^*, \mathbf{M}_1^*)$ is a basis of $\mathsf{span}(\mathbf{M}^\perp)$. □

The proof of Lemma 6 follows readily from Lemma 7, 8, 9, and 10. □

Lemma 11 ($\mathrm{G}_{2.\lambda}$). $\mathbf{Adv}_{2.\lambda} \leq \frac{Q_{\mathsf{enc}}}{q}$.

Proof of Lemma 11. We show that the joint distribution of all the values K_0 computed by EncO is statistically close to uniform over $\mathbb{G}^{Q_{\mathsf{enc}}}$. Recall that on input τ, $\mathsf{EncO}(\tau)$ computes

$$K_0 := [\mathbf{y}^\top(\mathbf{k}_\tau + \mathbf{M}^\perp \mathsf{RF}_\lambda(\tau))],$$

where $\mathsf{RF}_\lambda : \{0,1\}^\lambda \to \mathbb{Z}_q^{2k}$ is a random function, and $\mathbf{y} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k}$ (see Figure 6).

We make use of the following properties:

Property 1: all the tags τ queried to EncO, such that $\mathsf{EncO}(\tau) \neq \perp$, are distinct.

Property 2: the outputs of DecO are independent of $\{\mathsf{RF}(\tau) : \tau \in \mathcal{T}_{\mathsf{enc}}\}$. This is because for all queries $(\tau, [\mathbf{y}], \widehat{K})$ to DecO such that $\tau \in \mathcal{T}_{\mathsf{enc}}$, $\mathsf{DecO}(\tau, [\mathbf{y}], \widehat{K}) = 0$, independently of $\mathsf{RF}_\lambda(\tau)$, by definition of $\mathrm{G}_{2.\lambda}$.

Property 3: with probability at least $1 - \frac{Q_{\mathsf{enc}}}{q}$ over the random coins of EncO, all the vectors $\mathbf{y}$ sampled by EncO are such that $\mathbf{y}^\top \mathbf{M}^\perp \neq \mathbf{0}$.

We deduce that the joint distribution of all the values $\mathsf{RF}_\lambda(\tau)$ computed by EncO is uniformly random over $\left(\mathbb{Z}_q^{2k}\right)^{Q_{\mathsf{enc}}}$ (from Property 1), independent of the outputs of DecO (from Property 2). Finally, from Property 3, we get that the joint distribution of all the values K_0 computed by EncO is statistically close to uniform over $\mathbb{G}^{Q_{\mathsf{enc}}}$, since:

$$K_0 := [\mathbf{y}^\top(\mathbf{k}_\tau + \mathbf{M}^\perp \mathsf{RF}_\lambda(\tau)) = [\mathbf{y}^\top \mathbf{k}_\tau + \underbrace{\mathbf{y}^\top \mathbf{M}^\perp}_{\neq \mathbf{0} \text{ w.h.p.}} \mathsf{RF}_\lambda(\tau)].$$

This means that the values K_0 and K_1 are statistically close, and therefore, $\mathbf{Adv}_3 \leq \frac{Q_{\mathsf{enc}}}{q}$. □

Finally, Theorem 1 follows readily from Lemmas 4, 5, 6, and 11. □

4 Multi-ciphertext CCA-secure Public Key Encryption scheme

4.1 Our construction

We now describe the optimized IND-CCA-secure PKE scheme. Compared to the PCA-secure KEM from Section 3, we add an authenticated (symmetric) encryption scheme $(\mathsf{Enc}_{\mathsf{AE}}, \mathsf{Dec}_{\mathsf{AE}})$, and set the KEM tag τ as the hash value of a suitable part of the KEM ciphertext (as explained in the introduction). A formal definition with highlighted differences to our PCA-secure KEM appears in Figure 8.

We prove the security under the $\mathcal{U}_k$-MDDH Assumption, which admits a tight reduction to the standard k-Lin Assumption.

Theorem 2. *The Public Key Encryption scheme* $\mathsf{PKE}_{\mathsf{CCA}}$ *defined in Figure 8 has perfect correctness, if the underlying Authenticated Encryption scheme* AE *has perfect correctness. Moreover, if the* $\mathcal{U}_k$-MDDH *Assumption holds in* $\mathbb{G}$*,* AE *has one-time privacy and authenticity, and* $\mathcal{H}$ *generates collision resistant hash functions, then* $\mathsf{PKE}_{\mathsf{CCA}}$ *is* IND-CCA *secure. Namely, for any adversary* $\mathcal{A}$*, there exist adversaries* $\mathcal{B}$*,* $\mathcal{B}'$*,* $\mathcal{B}''$ *such that* $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{B}') \approx \mathbf{T}(\mathcal{B}'') \approx \mathbf{T}(\mathcal{A}) + (Q_{\mathsf{dec}} + Q_{\mathsf{enc}}) \cdot \mathsf{poly}(\lambda)$ *and*

$$\begin{aligned}\mathbf{Adv}^{\mathsf{ind\text{-}cca}}_{\mathsf{PKE}_{\mathsf{CCA}}}(\mathcal{A}) \leq (4\lambda+1) \cdot \mathbf{Adv}^{\mathrm{mddh}}_{\mathcal{U}_k, \mathsf{GGen}}(\mathcal{B}) + ((4\lambda+2)Q_{\mathsf{dec}} + Q_{\mathsf{enc}}) \cdot \mathbf{Adv}^{\mathsf{ae\text{-}ot}}_{\mathsf{AE}}(\mathcal{B}'') \\ + \mathbf{Adv}^{\mathrm{cr}}_{\mathcal{H}}(\mathcal{B}') + Q_{\mathsf{enc}}(Q_{\mathsf{enc}} + Q_{\mathsf{dec}}) \cdot 2^{-\Omega(\lambda)},\end{aligned} \tag{3}$$

$\underline{\mathsf{Gen}_{\mathsf{PKE}}(1^\lambda)}$:
$\mathcal{G} \leftarrow_{\mathrm{R}} \mathsf{GGen}(1^\lambda)$; $\mathsf{H} \leftarrow_{\mathrm{R}} \mathcal{H}(1^\lambda)$; $\mathbf{M} \leftarrow_{\mathrm{R}} \mathcal{U}_{3k,k}$
$\mathbf{k}_{1,0}, \ldots, \mathbf{k}_{\lambda,1} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k}$
$\mathsf{pk} := \left(\mathcal{G}, [\mathbf{M}], \mathsf{H}, \left([\mathbf{M}^\top \mathbf{k}_{j,\beta}]\right)_{1 \le j \le \lambda, 0 \le \beta \le 1}\right)$
$\mathsf{sk} := (\mathbf{k}_{j,\beta})_{1 \le j \le \lambda, 0 \le \beta \le 1}$
Return $(\mathsf{pk}, \mathsf{sk})$

$\underline{\mathsf{Enc}_{\mathsf{PKE}}(\mathsf{pk}, M)}$:
$\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$; $\mathbf{y} := \mathbf{M}\mathbf{r}$
$\tau := \mathsf{H}([\overline{\mathbf{y}}])$
$\mathbf{k}_\tau := \sum_{j=1}^{\lambda} \mathbf{k}_{j,\tau_j}$
$K := [\mathbf{r}^\top \cdot \mathbf{M}^\top \mathbf{k}_\tau]$
$\phi := \mathsf{Enc}_{\mathsf{AE}}(K, M)$
Return $([\mathbf{y}], \phi)$

$\underline{\mathsf{Dec}_{\mathsf{PKE}}(\mathsf{pk}, \mathsf{sk}, ([\mathbf{y}], \phi))}$:
$\tau := \mathsf{H}([\overline{\mathbf{y}}])$; $\mathbf{k}_\tau := \sum_{j=1}^{\lambda} \mathbf{k}_{j,\tau_j}$; $K := [\mathbf{y}^\top \mathbf{k}_\tau]$
Return $\mathsf{Dec}_{\mathsf{AE}}(K, \phi)$.

Fig. 8. $\mathsf{PKE}_{\mathsf{CCA}}$, an IND-CCA-secure PKE. We color in blue the differences with $\mathsf{KEM}_{\mathsf{PCA}}$, the IND-PCA-secure KEM in Figure 4. Here, GGen is a prime-order group generator (see Section 2.3) , and $\mathsf{AE} := (\mathsf{Enc}_{\mathsf{AE}}, \mathsf{Dec}_{\mathsf{AE}})$ is an Authenticated Encryption scheme with key-space $\mathcal{K} := \mathcal{G}$ (see Definition 7).

where Q_{enc}, Q_{dec} are the number of times $\mathcal{A}$ queries EncO, DecO, respectively, and $\mathsf{poly}(\lambda)$ is independent of $\mathbf{T}(\mathcal{A})$.

We note that the Q_{enc} and Q_{dec} factors in (3) are only related to AE. Hence, when using a statistically secure (one-time) authenticated encryption scheme, the corresponding terms in (3) become exponentially small.

Remark 2 (Extension to the multi-user CCA security). We only provide an analysis in the multi-ciphertext (but single-user) setting. However, we remark (without proof) that our analysis generalizes to the multi-user, multi-ciphertext scenario, similar to [6, 17, 19]. Indeed, all computational steps (not counting the steps related to the AE scheme) modify all ciphertexts simultaneously, relying for this on the re-randomizability of the $\mathcal{U}_k$-MDDH Assumption relative to a fixed matrix $\mathbf{M}$. The same modifications can be made to many $\mathsf{PKE}_{\mathsf{CCA}}$ simultaneously by using that the $\mathcal{U}_k$-MDDH Assumption is also re-randomizable across many matrices $\mathbf{M}_i$. (A similar property for the DDH, DLIN, and bilinear DDH assumptions is used in [6], [17], and [19], respectively.)

5 Security proof of $\mathsf{PKE}_{\mathsf{CCA}}$

Theorem 3. *The Public Key Encryption scheme $\mathsf{PKE}_{\mathsf{CCA}}$ defined in Figure 8, Section 3 has perfect correctness, provided the underlying Authenticated Encryption scheme AE has perfect correctness. Moreover, if the $\mathcal{U}_k$-MDDH Assumption holds in $\mathbb{G}$, AE has one-time privacy and authenticity, and $\mathcal{H}$ generates collision resistant hash functions, then $\mathsf{PKE}_{\mathsf{CCA}}$ is* IND-CCA *secure. Namely, for any adversary $\mathcal{A}$, there exist adversaries $\mathcal{B}$, $\mathcal{B}'$, $\mathcal{B}''$ such that $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{B}') \approx \mathbf{T}(\mathcal{B}'') \approx \mathbf{T}(\mathcal{A}) + (Q_{\mathsf{dec}} + Q_{\mathsf{enc}}) \cdot \mathsf{poly}(\lambda)$ and*

$$\begin{aligned}\mathbf{Adv}^{\mathsf{ind\text{-}cca}}_{\mathsf{PKE}_{\mathsf{CCA}}}(\mathcal{A}) \le{}& (4\lambda+1) \cdot \mathbf{Adv}^{\mathrm{mddh}}_{\mathcal{U}_k,\mathsf{GGen}}(\mathcal{B}) + (Q_{\mathsf{enc}}Q_{\mathsf{dec}} + (4\lambda+2)Q_{\mathsf{dec}} + Q_{\mathsf{enc}}) \cdot \mathbf{Adv}^{\mathsf{ae\text{-}ot}}_{\mathsf{AE}}(\mathcal{B}'') \\ &+ \mathbf{Adv}^{\mathrm{cr}}_{\mathcal{H}}(\mathcal{B}') + Q_{\mathsf{enc}}(Q_{\mathsf{enc}} + Q_{\mathsf{dec}}) \cdot 2^{-\Omega(\lambda)},\end{aligned} \tag{4}$$

where Q_{enc}, Q_{dec} are the number of times $\mathcal{A}$ queries EncO, DecO, respectively, and $\mathsf{poly}(\lambda)$ is independent of $\mathbf{T}(\mathcal{A})$.

We note that the Q_{enc} and Q_{dec} factors in (4) are only related to AE. Hence, when using a statistically secure authenticated encryption scheme, the corresponding terms in (4) become exponentially small.

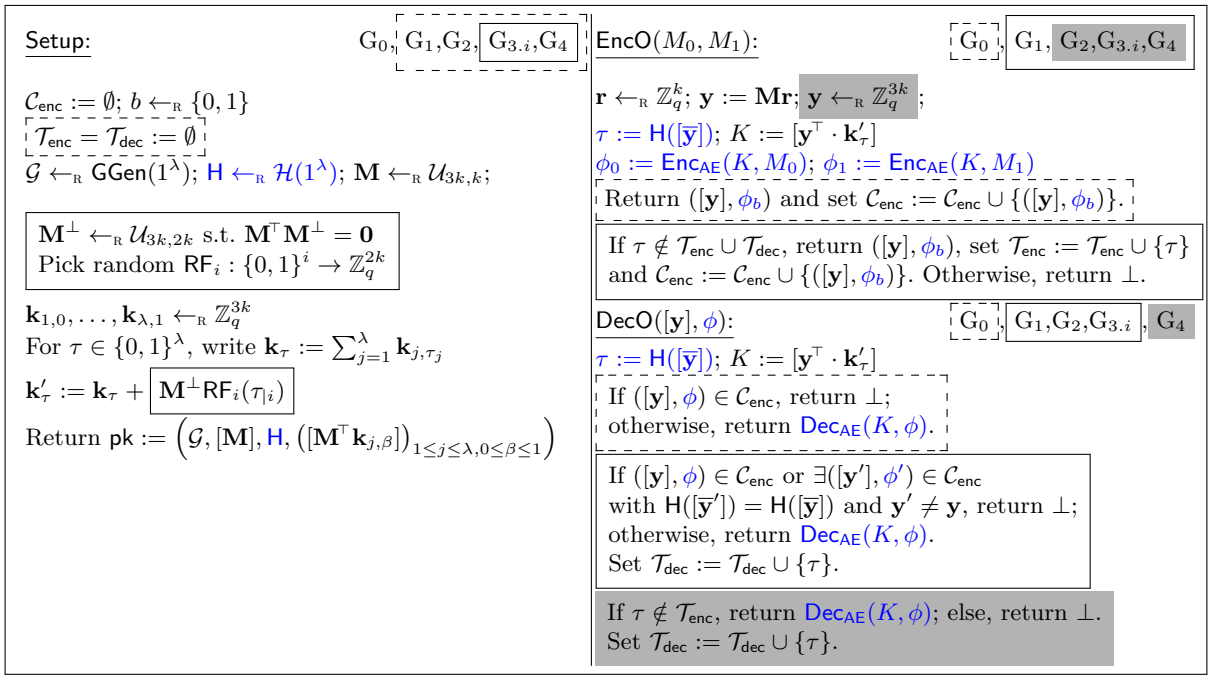

Fig. 9. Games $\mathrm{G}_0, \mathrm{G}_1, \mathrm{G}_2, \mathrm{G}_{3.i}$ (for $0 \le i \le \lambda$), G_4 for the proof of multi-ciphertext CCA security of $\mathsf{PKE}_{\mathsf{CCA}}$ in Figure 8. In each procedure, the components inside a solid (dotted, gray) frame are only present in the games marked by a solid (dotted, gray) frame. We color in blue the differences with Figure 6, for the security proof of $\mathsf{KEM}_{\mathsf{PCA}}$.

Proof of Theorem 3. Perfect correctness follows from the perfect correctness of AE and the fact that for all $\mathbf{r} \in \mathbb{Z}_q^k$ and $\mathbf{y} = \mathbf{M}\mathbf{r}$, for all $\mathbf{k} \in \mathbb{Z}_q^{3k}$:

$$\mathbf{r}^\top (\mathbf{M}^\top \mathbf{k}) = \mathbf{y}^\top \cdot \mathbf{k}.$$

We now prove the IND-CCA security of $\mathsf{PKE}_{\mathsf{CCA}}$. We proceed via a series of games described in Figures 9 and 10 and we use $\mathbf{Adv}_i$ to denote the advantage of $\mathcal{A}$ in game G_i.

Lemma 12 (G_0 **to** G_1)**.** *There exist adversaries $\mathcal{B}_0$ and $\mathcal{B}'_0$ such that $\mathbf{T}(\mathcal{B}_0) \approx \mathbf{T}(\mathcal{B}'_0) \approx \mathbf{T}(\mathcal{A}) + (Q_{\mathsf{enc}} + Q_{\mathsf{dec}}) \cdot \mathsf{poly}(\lambda)$ and*

$$|\mathbf{Adv}_0 - \mathbf{Adv}_1| = 2Q_{\mathsf{dec}} \cdot \mathbf{Adv}^{\mathsf{ae\text{-}ot}}_{\mathsf{AE}}(\mathcal{B}_0) + \mathbf{Adv}^{\mathrm{cr}}_{\mathcal{H}}(\mathcal{B}'_0) + \frac{Q_{\mathsf{enc}}(Q_{\mathsf{enc}} + Q_{\mathsf{dec}})}{q^k},$$

where Q_{enc}, Q_{dec} are the number of times $\mathcal{A}$ queries EncO, DecO, respectively, and $\mathsf{poly}(\lambda)$ is independent of $\mathbf{T}(\mathcal{A})$.

Here, we use the collision resistance of $\mathcal{H}$ and the one-time authenticity of AE to restrict the oracles DecO and EncO.

Proof of Lemma 12. First, we use the one-time authenticity of AE to argue that if $\mathcal{A}$ queries DecO on a vector $[\mathbf{y}]$ such that $\mathbf{y} \notin \mathsf{span}(\mathbf{M})$, then, DecO outputs $\perp$, with overwhelming probability over

the random coins of Setup. Second, we use the collision resistance of H to argue that:
(i) if $\mathcal{A}$ queries DecO on $([\mathbf{y}'], \phi')$, where for some previous output $([\mathbf{y}], \phi)$ of EncO, we have: $\mathsf{H}([\overline{\mathbf{y}}]) = \mathsf{H}([\overline{\mathbf{y}'}])$ and $\mathbf{y}' \neq \mathbf{y}$, then, with overwhelming probability over the random coins of $\mathcal{A}$, Setup and EncO: DecO outputs $\perp$;
(ii) every time EncO outputs a vector $[\mathbf{y}]$, its tag $\mathsf{H}([\overline{\mathbf{y}}])$ is fresh (no $[\mathbf{y}']$ with the same tag has been output by EncO or queried to DecO before), with overwhelming probability over EncO's random coins.

We introduce intermediate games $\mathrm{G}_{0.j}$ (resp. $\mathrm{G}_{1.j}$) for $j = 0, \ldots, Q_{\mathsf{dec}}$, defined as follows: DecO is as in G_0 (resp. G_1) except that for the first j times it is queried, it outputs $\perp$ to any query $([\mathbf{y}], \phi)$ such that $\mathbf{y} \notin \mathsf{span}(\mathbf{M})$. Setup and EncO are as in G_0 (resp. G_1).

We show that:

$$\mathrm{G}_0 \equiv \mathrm{G}_{0.0} \approx_{\mathsf{AE}} \mathrm{G}_{0.1} \approx_{\mathsf{AE}} \ldots \approx_{\mathsf{AE}} \mathrm{G}_{0.Q_{\mathsf{dec}}} \approx_{CR} \mathrm{G}_{1.Q_{\mathsf{dec}}} \approx_{\mathsf{AE}} \ldots \approx_{\mathsf{AE}} \mathrm{G}_{1.0} \equiv \mathrm{G}_1$$

where $\equiv$ denotes statistical equality, $\approx_{\mathsf{AE}}$ denotes indistinguishability based on the security of AE, and $\approx_{CR}$ denotes indistinguishability based on the collision resistance of $\mathcal{H}$.

Namely, we build adversaries $\mathcal{B}_{0.j}$, $\mathcal{B}_{1.j}$ for $j = 0, \ldots, Q_{\mathsf{dec}} - 1$, and $\mathcal{B}'_0$ such that $\mathbf{T}(\mathcal{B}_{0,j}) \approx \mathbf{T}(\mathcal{B}_{1,j}) \approx \mathbf{T}(\mathcal{B}'_0) \approx \mathbf{T}(\mathcal{A}) + (Q_{\mathsf{enc}} + Q_{\mathsf{dec}}) \cdot \mathsf{poly}(\lambda)$, where $\mathsf{poly}(\lambda)$ is independent of $\mathbf{T}(\mathcal{A})$, and such that

Claim 1: $|\mathbf{Adv}_{0.j} - \mathbf{Adv}_{0.j+1}| \leq \mathbf{Adv}^{\mathsf{ae\text{-}ot}}_{\mathsf{AE}}(\mathcal{B}_{0.j})$ and $|\mathbf{Adv}_{1.j} - \mathbf{Adv}_{1.j+1}| \leq \mathbf{Adv}^{\mathsf{ae\text{-}ot}}_{\mathsf{AE}}(\mathcal{B}_{0.j})$, for $j = 0, \ldots, Q_{\mathsf{dec}} - 1$.
Claim 2: $|\mathbf{Adv}_{0.Q_{\mathsf{dec}}} - \mathbf{Adv}_{1.Q_{\mathsf{dec}}}| \leq \mathbf{Adv}^{\mathrm{cr}}_{\mathcal{H}}(\mathcal{B}'_0)$.

This implies the lemma.

Let us prove Claim 1. It suffices to show that in $\mathrm{G}_{0.j}$ and $\mathrm{G}_{1.j}$, with overwhelming probability over the random coins of Setup, DecO outputs $\perp$ to its $j+1$-st query if it contains $[\mathbf{y}]$ such that $\mathbf{y} \notin \mathsf{span}(\mathbf{M})$.

Recall that in both $\mathrm{G}_{0.j}$ and $\mathrm{G}_{1.j}$, on its $j+1$-st query $([\mathbf{y}], \phi)$, DecO computes

$$K := [\mathbf{y}^\top \cdot \mathbf{k}_\tau], \text{ where } \tau = \mathsf{H}([\overline{\mathbf{y}}]) \text{ and } \mathbf{k}_\tau := \sum_{\rho=1}^{\lambda} \mathbf{k}_{\rho,\tau_\rho},$$

and returns $\mathsf{Dec}_{\mathsf{AE}}(K, \phi)$ (or $\perp$, see Figure 9). We prove that this value K is hidden from $\mathcal{A}$ up to its $j+1$-st query to DecO. Then, we use the one-time authenticity of AE to argue that $\mathsf{Dec}_{\mathsf{AE}}(K, \phi) = \perp$ with overwhelming probability.

To prove K is hidden from $\mathcal{A}$, we show that the vectors $\mathbf{k}_{1,0}, \mathbf{k}_{1,1}$ in sk contain some entropy that is hidden from $\mathcal{A}$. More formally, we use the fact that the vectors $\mathbf{k}_{1,\beta} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k}$ are identically distributed than $\mathbf{k}_{1,\beta} + \mathbf{M}^\perp \mathbf{w}$ for $\beta = 0, 1$, where $\mathbf{k}_{1,\beta} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k}$, $\mathbf{w} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$, and $\mathbf{M}^\perp \leftarrow_{\mathrm{R}} \mathcal{U}_{3k,2k}$ such that $\mathbf{M}^\top \mathbf{M}^\perp = \mathbf{0}$. We show that $\mathbf{w}$ is hidden from $\mathcal{A}$, up to its $j+1$-st query to DecO.

– The public key pk does not leak any information about $\mathbf{w}$, since

$$\mathbf{M}^\top (\mathbf{k}_{1,\beta} + \boxed{\mathbf{M}^\perp \mathbf{w}}) = \mathbf{M}^\top \mathbf{k}_{1,\beta}.$$

This is because $\mathbf{M}^\top \mathbf{M}^\perp = \mathbf{0}$.

– The outputs of EncO also hide $\mathbf{w}$, since for any $\mathbf{y} \in \mathsf{span}(\mathbf{M})$, we have:

$$\mathbf{y}^\top(\mathbf{k}_\tau + \boxed{\mathbf{M}^\perp \mathbf{w}}) = \mathbf{y}^\top \mathbf{k}'_\tau \tag{5}$$

since $\mathbf{M}^\top \mathbf{M}^\perp = \mathbf{0}$ which implies $\mathbf{y}^\top \mathbf{M}^\perp = \mathbf{0}$.

– The first j outputs of DecO also hide $\mathbf{w}$.
  - For $\mathbf{y} \in \mathsf{span}(\mathbf{M})$, $\mathsf{DecO}([\mathbf{y}], \phi)$ is independent of $\mathbf{w}$, from Equation (5).
  - For $\mathbf{y} \notin \mathsf{span}(\mathbf{M})$, $\mathsf{DecO}([\mathbf{y}], \phi) = \perp$, independently of $\mathbf{w}$, by definition of $\mathrm{G}_{0.j}$.

Therefore, the value

$$K = [\mathbf{y}^\top(\mathbf{k}_\tau + \mathbf{M}^\perp \mathbf{w})] = [\mathbf{y}^\top \mathbf{k}_\tau + \underbrace{\mathbf{y}^\top \mathbf{M}^\perp}_{\neq \mathbf{0}} \mathbf{w}]$$

computed by DecO on its $j+1$-st query, is uniformly random over $\mathbb{G}$ from $\mathcal{A}$'s view, since $\mathbf{y} \notin \mathsf{span}(\mathbf{M}) \Leftrightarrow \mathbf{y}^\top \mathbf{M}^\perp \neq \mathbf{0}$.

Then, by one-time authenticity of AE, there exists an adversary $\mathcal{B}_{0.j}$ such that $\mathbf{T}(\mathcal{B}_{0,j}) \approx \mathbf{T}(\mathcal{A}) + (Q_{\mathsf{enc}} + Q_{\mathsf{dec}}) \cdot \mathsf{poly}(\lambda)$, where $\mathsf{poly}(\lambda)$ is independent of $\mathbf{T}(\mathcal{A})$, and

$$|\mathbf{Adv}_{0.j} - \mathbf{Adv}_{0.j+1}| \leq \mathbf{Adv}^{\mathsf{ae\text{-}ot}}_{\mathsf{AE}}(\mathcal{B}_{0.j}).$$

Let us prove Claim 2. It suffices to show that in $\mathrm{G}_{0.Q_{\mathsf{dec}}}$:
(i) if DecO is queried on $([\mathbf{y}], \phi)$, and there exists $([\mathbf{y}'], \phi')$ output previously by EncO, with $\mathsf{H}([\overline{\mathbf{y}}]) = \mathsf{H}([\overline{\mathbf{y}'}])$ and $\mathbf{y}' \neq \mathbf{y}$, then, with overwhelming probability over the random coins of $\mathcal{A}$, Setup and EncO: DecO outputs $\perp$;
(ii) every time EncO outputs a vector $[\mathbf{y}]$, its tag $\mathsf{H}([\overline{\mathbf{y}}])$ is fresh (no $[\mathbf{y}']$ with the same tag has been output by EncO or queried to DecO before), with overwhelming probability over its random coins.

We define $\mathcal{B}'_0$ as follows. Upon receiving a challenge $\mathsf{H} \leftarrow_{\mathrm{R}} \mathcal{H}(1^\lambda)$ for the collision resistance of $\mathcal{H}$, $\mathcal{B}'_0$ picks $b \leftarrow_{\mathrm{R}} \{0,1\}$, $\mathbf{k}_{1,0}, \ldots, \mathbf{k}_{\lambda,1} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k}$, and simulates Setup, EncO and DecO as in $\mathrm{G}_{0.Q_{\mathsf{dec}}}$.

(i) Suppose $\mathcal{B}'_0$ receives some $[\mathbf{y}]$ through a DecO query, such that there is a $[\mathbf{y}']$ from an earlier EncO query with $\mathsf{H}([\overline{\mathbf{y}}]) = \mathsf{H}([\overline{\mathbf{y}'}])$, and $\mathbf{y} \neq \mathbf{y}'$. Then, we distinguish the following cases:

Case 1: $\overline{\mathbf{y}} \neq \overline{\mathbf{y}}'$. Then there is a collision $\mathsf{H}([\overline{\mathbf{y}}]) = \mathsf{H}([\overline{\mathbf{y}}'])$ that $\mathcal{B}'_0$ can directly output.
Case 2: $\overline{\mathbf{y}} = \overline{\mathbf{y}}'$ (but $\mathbf{y} \neq \mathbf{y}'$). Then, $\mathbf{y} \notin \mathsf{span}(\mathbf{M})$ (because $\mathbf{y} \neq \mathbf{y}'$), and DecO outputs $\perp$, as would happen both in $\mathrm{G}_{0.Q_{\mathsf{dec}}}$ and $\mathrm{G}_{1.Q_{\mathsf{dec}}}$.

(ii) First, note that with probability at least $1 - \frac{Q_{\mathsf{enc}}(Q_{\mathsf{enc}}+Q_{\mathsf{dec}})}{q^k}$ over its random coins, EncO samples vectors $[\mathbf{y}]$ whose upper parts $[\overline{\mathbf{y}}]$ are fresh (they are distinct from those previously sampled by EncO, or queried to DecO). Therefore, conditioned on this fact, if $\mathcal{B}'_0$ samples $\tau := \mathsf{H}([\overline{\mathbf{y}}])$ that is not fresh, i.e there exists a pair $([\mathbf{y}'], \mathsf{H}([\overline{\mathbf{y}}']) = \tau)$ previously output by EncO or queried to DecO (along with some symmetric ciphertext ϕ), then we have $\mathsf{H}([\overline{\mathbf{y}}]) = \mathsf{H}([\overline{\mathbf{y}}'])$, and $[\overline{\mathbf{y}}] \neq [\overline{\mathbf{y}}']$, that is, $\mathcal{B}'_0$ finds a collision.

Summarizing, both games $\mathrm{G}_{0.Q_{\mathsf{dec}}}$ and $\mathrm{G}_{1.Q_{\mathsf{dec}}}$ proceed identically (as simulated by $\mathcal{B}'_0$), unless (i) Case 1 occurs, or (ii) EncO samples a tag that was output or queried before, in which case $\mathcal{B}'_0$ finds a collision, with overwhelming probability over its random coins. □

Lemma 13 (G_1 to G_2). *There exists an adversary $\mathcal{B}_1$ such that $\mathbf{T}(\mathcal{B}_1) \approx \mathbf{T}(\mathcal{A}) + (Q_{\mathsf{enc}} + Q_{\mathsf{dec}}) \cdot \mathsf{poly}(\lambda)$ and*

$$|\mathbf{Adv}_1 - \mathbf{Adv}_2| = \mathbf{Adv}^{\mathrm{mddh}}_{\mathcal{U}_k, \mathsf{GGen}}(\mathcal{B}_1) + \frac{1}{q-1},$$

where Q_{enc}, Q_{dec} *are the number of times* $\mathcal{A}$ *queries* EncO, DecO, *respectively, and* $\mathsf{poly}(\lambda)$ *is independent of* $\mathbf{T}(\mathcal{A})$.

In Lemma 13, we use the MDDH assumption to "tightly" switch the distributions of all the challenge ciphertexts, as for Lemma 4 in Section 3.

Lemma 14 (G_2 **to** $\mathrm{G}_{3.0}$)**.** $|\mathbf{Adv}_2 - \mathbf{Adv}_{3.0}| = 0$.

The proofs of Lemma 13 and 14 are almost identical to those of Lemma 4 and 5, respectively. See the latter for further details.

Lemma 15 ($\mathrm{G}_{3.i}$ **to** $\mathrm{G}_{3.i+1}$)**.** *For all* $0 \leq i \leq \lambda - 1$, *there exist adversaries* $\mathcal{B}_{3.i}$ *and* $\mathcal{B}'_{3.i}$ *such that* $\mathbf{T}(\mathcal{B}_{3.i}) \approx \mathbf{T}(\mathcal{B}'_{3.i}) \approx \mathbf{T}(\mathcal{A}) + (Q_{\mathsf{enc}} + Q_{\mathsf{dec}}) \cdot \mathsf{poly}(\lambda)$ *and*

$$|\mathbf{Adv}_{3.i} - \mathbf{Adv}_{3.i+1}| \leq 4 \cdot \mathbf{Adv}^{\mathrm{mddh}}_{\mathcal{U}_k,\mathsf{GGen}}(\mathcal{B}_{3.i}) + 4Q_{\mathsf{dec}} \cdot \mathbf{Adv}^{\mathsf{ae\text{-}ot}}_{\mathsf{AE}}(\mathcal{B}'_{3.i}) + \frac{4}{q-1} + \frac{2k}{q},$$

where Q_{enc}, Q_{dec} *are the number of times* $\mathcal{A}$ *queries* EncO, DecO, *respectively, and* $\mathsf{poly}(\lambda)$ *is independent of* $\mathbf{T}(\mathcal{A})$.

Proof of Lemma 15. To go from $\mathrm{G}_{3.i}$ to $\mathrm{G}_{3.i+1}$, we introduce intermediate games $\mathrm{G}_{3.i.1}$, $\mathrm{G}_{3.i.2}$ and $\mathrm{G}_{3.i.3}$, defined in Figure 10. We prove that these games are indistinguishable in Lemma 16, 17, 18, and 19.

Lemma 16 ($\mathrm{G}_{3.i}$ **to** $\mathrm{G}_{3.i.1}$)**.** *For all* $0 \leq i \leq \lambda - 1$, *there exists an adversary* $\mathcal{B}_{3.i.0}$ *such that* $\mathbf{T}(\mathcal{B}_{3.i.0}) \approx \mathbf{T}(\mathcal{A}) + (Q_{\mathsf{enc}} + Q_{\mathsf{dec}}) \cdot \mathsf{poly}(\lambda)$ *and*

$$|\mathbf{Adv}_{3.i} - \mathbf{Adv}_{3.i.1}| \leq 2 \cdot \mathbf{Adv}^{\mathrm{mddh}}_{\mathcal{U}_k,\mathsf{GGen}}(\mathcal{B}_{3.i.0}) + \frac{2}{q-1},$$

where $\mathsf{poly}(\lambda)$ *is independent of* $\mathbf{T}(\mathcal{A})$.

Here, we use the MDDH Assumption to "tightly" switch the distribution of all the challenge ciphertexts, as in Lemma 7 in Section 3. We proceed in two steps, first, by changing the distribution of all the ciphertexts with a tag τ such that $\tau_{i+1} = 0$, and then, for those with a tag τ such that $\tau_{i+1} = 1$. We use the MDDH Assumption with respect to an independent matrix for each step.
Proof of Lemma 16. The proof of this lemma is essentially as the proof of Lemma 7, in Section 3. The difference is that now, only the lower part of the vectors $[\mathbf{y}]$ sampled by EncO is randomized using the Q_{enc}-fold $\mathcal{U}_{2k,k}$-MDDH Assumption. The upper part of $[\mathbf{y}]$ is used to compute the tag τ. We call $\overline{\mathbf{y}}$ and $\underline{\mathbf{y}}$ the upper and lower part of $\mathbf{y}$, respectively.

We introduce an intermediate game $\mathrm{G}_{3.i.0}$ where EncO first picks $\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$, computes $[\overline{\mathbf{y}}] := [\overline{\mathbf{M}}\mathbf{r}]$, $\tau := \mathsf{H}([\overline{\mathbf{y}}])$, and computes the rest of its output as in $\mathrm{G}_{3.i.1}$ if $\tau_{i+1} = 0$, and as in $\mathrm{G}_{3.i}$ if $\tau_{i+1} = 1$; Setup and DecO are as in $\mathrm{G}_{3.i.1}$. We build adversaries $\mathcal{B}'_{3.i.0}$ and $\mathcal{B}''_{3.i.0}$ such that $\mathbf{T}(\mathcal{B}'_{3.i.0}) \approx \mathbf{T}(\mathcal{B}''_{3.i.0}) \approx \mathbf{T}(\mathcal{A}) + (Q_{\mathsf{enc}} + Q_{\mathsf{dec}}) \cdot \mathsf{poly}(\lambda)$ with $\mathsf{poly}(\lambda)$ independent of $\mathbf{T}(\mathcal{A})$, and

Claim 1: $|\mathbf{Adv}_{3.i} - \mathbf{Adv}_{3.i.0}| \leq \mathbf{Adv}^{\mathrm{Q_{enc}\text{-}mddh}}_{\mathcal{U}_{2k,k},\mathsf{GGen}}(\mathcal{B}'_{3.i.0})$.
Claim 2: $|\mathbf{Adv}_{3.i.0} - \mathbf{Adv}_{3.i.1}| \leq \mathbf{Adv}^{\mathrm{Q_{enc}\text{-}mddh}}_{\mathcal{U}_{2k,k},\mathsf{GGen}}(\mathcal{B}''_{3.i.0})$.

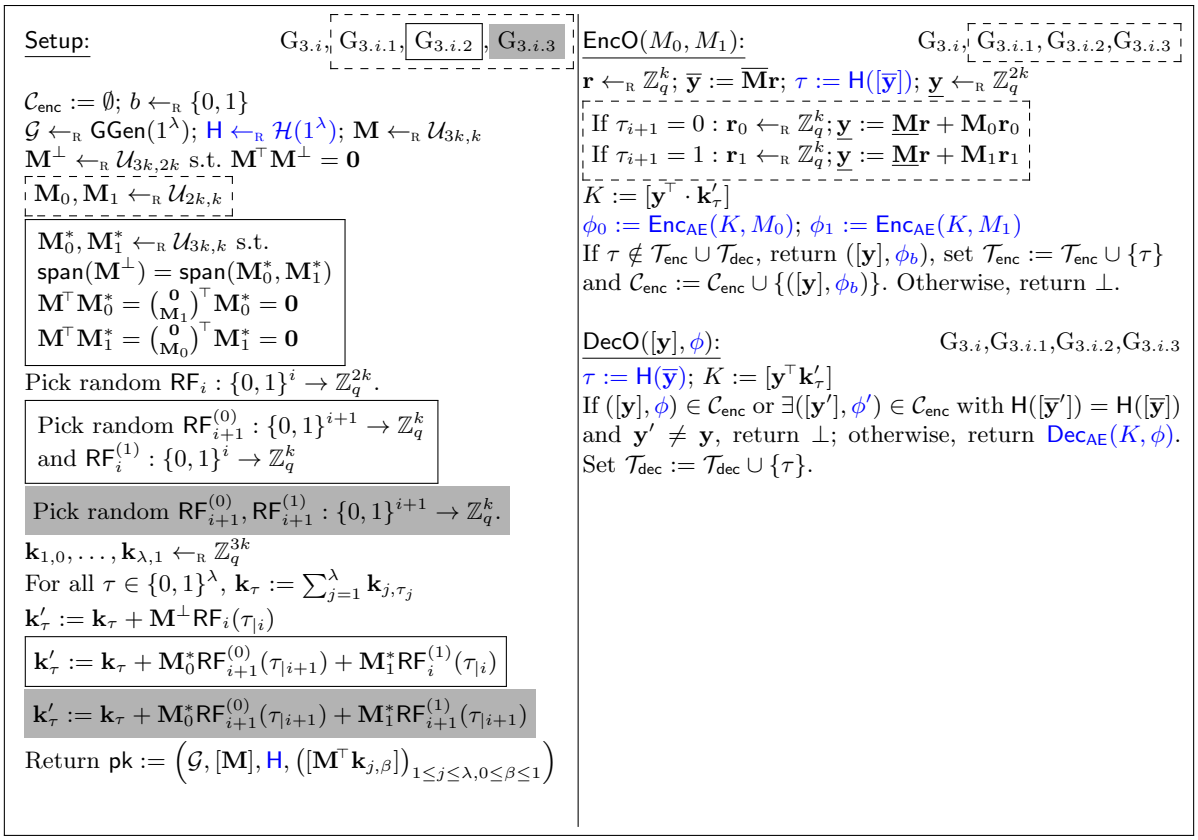

Fig. 10. Games $\mathrm{G}_{3.i}$ (for $0 \le i \le \lambda$),$\mathrm{G}_{3.i.1}$, $\mathrm{G}_{3.i.2}$ and $\mathrm{G}_{3.i.3}$ (for $0 \le i \le \lambda - 1$) for the proof of Lemma 15. For all $\tau \in \{0,1\}^\lambda$, we denote by $\tau_{|i}$ the i-bit prefix of τ. In each procedure, the components inside a solid (dotted, gray) frame are only present in the games marked by a solid (dotted, gray) frame. We color in blue the differences with Figure 7, for the security proof of $\mathsf{KEM}_{\mathsf{PCA}}$.

This implies the lemma by Lemma 3 (self-reducibility of $\mathcal{U}_{2k,k}$-MDDH), and Lemma 1 ($\mathcal{U}_{2k,k}$-MDDH $\Leftrightarrow \mathcal{U}_k$-MDDH).

Let us prove Claim 1. Upon receiving a challenge $(\mathcal{G}, [\mathbf{M}_0] \in \mathbb{G}^{2k \times k}, [\mathbf{H}] := [\mathbf{h}_1 | \ldots | \mathbf{h}_{Q_{\mathsf{enc}}}] \in \mathbb{G}^{2k \times Q_{\mathsf{enc}}})$ for the Q_{enc}-fold $\mathcal{U}_{2k,k}$-MDDH Assumption with respect to $\mathbf{M}_0 \leftarrow_{\mathrm{R}} \mathcal{U}_{2k,k}$, $\mathcal{B}'_{3.i.0}$ does as follows:

Setup: $\mathcal{B}'_{3.i.0}$ picks $\mathbf{M} \leftarrow_{\mathrm{R}} \mathcal{U}_{3k,k}$, $\mathbf{k}_{1,0}, \ldots, \mathbf{k}_{\lambda,1} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k}$, $\mathsf{H} \leftarrow_{\mathrm{R}} \mathcal{H}(1^\lambda)$, and computes pk as described in Figure 10. For each τ computed while simulating EncO or DecO, $\mathcal{B}'_{3.i.0}$ computes on the fly $\mathsf{RF}_i(\tau_{|i})$, $\mathbf{k}'_\tau := \mathbf{k}_\tau + \mathbf{M}^\perp \mathsf{RF}_i(\tau_{|i})$, where $\mathsf{RF}_i : \{0,1\}^i \to \mathbb{Z}_q^{2k}$ is a random function, $\mathbf{k}_\tau := \sum_{j=1}^{\lambda} \mathbf{k}_{j,\tau_j}$, and $\tau_{|i}$ denotes the i-bit prefix of τ (see Figure 10). Note that $\mathcal{B}'_{3.i.0}$ can compute efficiently $\mathbf{M}^\perp$ from $\mathbf{M}$.

$\mathsf{EncO}(M_0, M_1)$: on the j'th query, for $j = 1, \ldots, Q_{\mathsf{enc}}$, $\mathcal{B}'_{3.i.0}$ samples $\mathbf{r} \leftarrow \mathbb{Z}_q^k$, computes $[\overline{\mathbf{y}}] := [\overline{\mathbf{M}}\mathbf{r}]$, $\tau := \mathsf{H}([\overline{\mathbf{y}}])$, and computes $[\underline{\mathbf{y}}]$ as follows:

$$\begin{array}{l} \text{if } \tau_{i+1} = 0 : [\underline{\mathbf{y}}] := [\underline{\mathbf{M}}\mathbf{r} + \mathbf{h}_j] \\ \text{if } \tau_{i+1} = 1 : [\underline{\mathbf{y}}] \leftarrow_{\mathrm{R}} \mathbb{G}^{2k} \end{array}$$

This way, $\mathcal{B}'_{3.i.0}$ simulates EncO as in $\mathrm{G}_{3.i.0}$ when $[\mathbf{h}_j] := [\mathbf{M}_0\mathbf{r}_0]$ with $\mathbf{r}_0 \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$, and as in $\mathrm{G}_{3.i}$ when $[\mathbf{h}_j] \leftarrow_{\mathrm{R}} \mathbb{G}^{2k}$.

$\mathsf{DecO}(C, \phi)$**:** Finally, $\mathcal{B}'_{3.i.0}$ simulates DecO as described in Figure 10.

Therefore, $|\mathbf{Adv}_{3.i} - \mathbf{Adv}_{3.i.0}| \leq \mathbf{Adv}^{\mathrm{Q}_{\mathsf{enc}}\text{-mddh}}_{\mathcal{U}_{2k,k},\mathsf{GGen}}(\mathcal{B}'_{3.i.0})$.

To prove Claim 2, we build an adversary $\mathcal{B}''_{3.i.0}$ against the Q_{enc}-fold $\mathcal{U}_{2k,k}$-MDDH Assumption with respect to a matrix $\mathbf{M}_1 \leftarrow_{\mathrm{R}} \mathcal{U}_{2k,k}$, independent from $\mathbf{M}_0$, similarly than $\mathcal{B}'_{3.i.0}$. □

Lemma 17 ($\mathrm{G}_{3.i.1}$ **to** $\mathrm{G}_{3.i.2}$)**.** *For all $0 \leq i \leq \lambda - 1$, there exists an adversary $\mathcal{B}_{3.i.1}$ such that $\mathbf{T}(\mathcal{B}_{3.i.1}) \approx \mathbf{T}(\mathcal{A}) + (Q_{\mathsf{enc}} + Q_{\mathsf{dec}}) \cdot \mathsf{poly}(\lambda)$, and*

$$|\mathbf{Adv}_{3.i.1} - \mathbf{Adv}_{3.i.2}| \leq 2Q_{\mathsf{dec}} \cdot \mathbf{Adv}^{\mathsf{ae\text{-}ot}}_{\mathsf{AE}}(\mathcal{B}_{3.i.1}) + \frac{2k}{q}$$

where Q_{enc}, Q_{dec} are the number of times $\mathcal{A}$ queries EncO, DecO, respectively, and $\mathsf{poly}(\lambda)$ is independent of $\mathbf{T}(\mathcal{A})$.

Here, we use a computational variant of the Cramer-Shoup information-theoretic argument to move from RF_i to RF_{i+1}, thereby increasing the entropy of $\mathbf{k}'_\tau$, as in Lemma 8, in Section 3. For the sake of readability, we proceed in two steps: in Lemma 17, we move from RF_i to an hybrid between RF_i and RF_{i+1}, and in Lemma 18, we move to RF_{i+1}.

Proof of Lemma 17. In $\mathrm{G}_{3.i.2}$, we decompose $\mathsf{span}(\mathbf{M}^\perp)$ into two spaces $\mathsf{span}(\mathbf{M}_0^*)$ and $\mathsf{span}(\mathbf{M}_1^*)$, and we increase the entropy of the vector $\mathbf{k}'_\tau$ computed by EncO and DecO. More precisely, the entropy of the components of $\mathbf{k}'_\tau$ that lie in $\mathsf{span}(\mathbf{M}_0^*)$ increases from $\mathrm{G}_{3.i.1}$ to $\mathrm{G}_{3.i.2}$. To argue that these two games are computationally indistinguishable, we use a Cramer-Shoup argument [11], together with the one-time authenticity of AE.

Let us first explain how the matrices $\mathbf{M}_0^*$ and $\mathbf{M}_1^*$ are sampled. Note that with probability $1 - \frac{2k}{q}$ over the random coins of Setup, $(\mathbf{M} \| \binom{\mathbf{0}}{\mathbf{M}_0} \| \binom{\mathbf{0}}{\mathbf{M}_1})$ forms a basis of $\mathbb{Z}_q^{3k}$. Therefore, we have $\mathsf{span}(\mathbf{M}^\perp) = \mathsf{Ker}(\mathbf{M}^\top) = \mathsf{Ker}\big((\mathbf{M} \| \binom{\mathbf{0}}{\mathbf{M}_1})^\top\big) \oplus \mathsf{Ker}\big((\mathbf{M} \| \binom{\mathbf{0}}{\mathbf{M}_0})^\top\big)$.

We pick uniformly $\mathbf{M}_0^*$ and $\mathbf{M}_1^*$ in $\mathbb{Z}_q^{3k \times k}$ that generates $\mathsf{Ker}\big((\mathbf{M} \| \binom{\mathbf{0}}{\mathbf{M}_1})^\top\big)$ and $\mathsf{Ker}\big((\mathbf{M} \| \binom{\mathbf{0}}{\mathbf{M}_0})^\top\big)$, respectively. This way, for all $\tau \in \{0,1\}^\lambda$, we can write

$$\mathbf{M}^\perp \mathsf{RF}_i(\tau_{|i}) := \mathbf{M}_0^* \mathsf{RF}_i^{(0)}(\tau_{|i}) + \mathbf{M}_1^* \mathsf{RF}_i^{(1)}(\tau_{|i}),$$

where $\mathsf{RF}_i^{(0)}$, $\mathsf{RF}_i^{(1)} : \{0,1\}^i \to \mathbb{Z}_q^k$ are independent random functions.

We define $\mathsf{RF}_{i+1}^{(0)} : \{0,1\}^{i+1} \to \mathbb{Z}_q^k$ as follows:

$$\mathsf{RF}_{i+1}^{(0)}(\tau_{|i+1}) := \begin{cases} \mathsf{RF}_i^{(0)}(\tau_{|i}) & \text{if } \tau_{i+1} = 0 \\ \mathsf{RF}_i^{(0)}(\tau_{|i}) + \mathsf{RF}'^{(0)}_i(\tau_{|i}) & \text{if } \tau_{i+1} = 1 \end{cases}$$

where $\mathsf{RF}'^{(0)}_i : \{0,1\}^i \to \mathbb{Z}_q^k$ is a random function independent from $\mathsf{RF}_i^{(0)}$. This way, $\mathsf{RF}_{i+1}^{(0)}$ is a random function.

We show that the outputs of EncO and DecO are computationally indistinguishable in $\mathrm{G}_{3.i.1}$ and $\mathrm{G}_{3.i.2}$. We decompose the proof in two cases (delimited with ■): the queries corresponding to a tag $\tau \in \{0,1\}^\lambda$ such that $\tau_{i+1} = 0$, and the queries corresponding to a tag τ such that $\tau_{i+1} = 1$.

Queries with $\tau_{i+1} = 0$:
The only difference between $\mathrm{G}_{3.i.1}$ and $\mathrm{G}_{3.i.2}$ is that Setup computes $\mathbf{k}'_\tau$ using the random function $\mathsf{RF}_i^{(0)}$ in $\mathrm{G}_{3.i.1}$, whereas it uses the random function $\mathsf{RF}_{i+1}^{(0)}$ in $\mathrm{G}_{3.i.2}$ (see Figure 10). Therefore, by definition of $\mathsf{RF}_{i+1}^{(0)}$, for all $\tau \in \{0,1\}^\lambda$ such that $\tau_{i+1} = 0$, $\mathbf{k}'_\tau$ is the same in $\mathrm{G}_{3.i.1}$ and $\mathrm{G}_{3.i.2}$, and the outputs of EncO and DecO are identically distributed. ■
Queries with $\tau_{i+1} = 1$:
Observe that for all $\mathbf{y} \in \mathsf{span}(\mathbf{M}, \binom{\mathbf{0}}{\mathbf{M}_1})$ and all $\tau \in \{0,1\}^\lambda$ such that $\tau_{i+1} = 1$,

$$\begin{aligned}
&\overbrace{\mathbf{y}^\top \Big(\mathbf{k}_\tau + \mathbf{M}_0^* \mathsf{RF}_i^{(0)}(\tau_{|i}) + \mathbf{M}_1^* \mathsf{RF}_i^{(1)}(\tau_{|i}) + \boxed{\mathbf{M}_0^* \mathsf{RF}'^{(0)}_i(\tau_{|i})}\Big)}^{\mathrm{G}_{3.i.2}} \\
&= \mathbf{y}^\top \Big(\mathbf{k}_\tau + \mathbf{M}_0^* \mathsf{RF}_i^{(0)}(\tau_{|i}) + \mathbf{M}_1^* \mathsf{RF}_i^{(1)}(\tau_{|i})\Big) + \underbrace{\boxed{\mathbf{y}^\top \mathbf{M}_0^* \mathsf{RF}'^{(0)}_i(\tau_{|i})}}_{=0} \\
&= \overbrace{\mathbf{y}^\top \cdot \Big(\mathbf{k}_\tau + \mathbf{M}_0^* \mathsf{RF}_i^{(0)}(\tau_{|i}) + \mathbf{M}_1^* \mathsf{RF}_i^{(1)}(\tau_{|i})\Big)}^{\mathrm{G}_{3.i.1}}
\end{aligned}$$

where the second equality uses the fact $\mathbf{M}^\top \mathbf{M}_0^* = \binom{\mathbf{0}}{\mathbf{M}_1}^\top \mathbf{M}_0^* = \mathbf{0}$ and thus $\mathbf{y}^\top \mathbf{M}_0^* = \mathbf{0}$.

This means that:

- the outputs of EncO that contains $[\mathbf{y}]$ whose tag $\tau = \mathsf{H}([\overline{\mathbf{y}}])$ is such that $\tau_{i+1} = 1$ are identically distributed in $\mathrm{G}_{3.i.1}$ and $\mathrm{G}_{3.i.2}$;
- the output of DecO on any input $([\mathbf{y}], \phi)$ where $\tau = \mathsf{H}([\overline{\mathbf{y}}])$, $\tau_{i+1} = 1$, and $\mathbf{y} \in \mathsf{span}(\mathbf{M}, \binom{\mathbf{0}}{\mathbf{M}_1})$ is the same in $\mathrm{G}_{3.i.1}$ and $\mathrm{G}_{3.i.2}$.

Henceforth, we focus on the *ill-formed* queries to DecO, namely those corresponding to $\tau_{i+1} = 1$, and $\mathbf{y} \notin \mathsf{span}(\mathbf{M}, \binom{\mathbf{0}}{\mathbf{M}_1})$. We introduce intermediate games $\mathrm{G}_{3.i.1.j}$, and $\mathrm{G}'_{3.i.1.j}$ for $j = 0, \ldots, Q_{\mathsf{dec}}$, defined as follows:

- $\mathrm{G}_{3.i.1.j}$: DecO is as in $\mathrm{G}_{3.i.1}$ except that for the first j times it is queried, it outputs $\perp$ to any ill-formed query. EncO is as in $\mathrm{G}_{3.i.2}$.
- $\mathrm{G}'_{3.i.1.j}$: DecO is as in $\mathrm{G}_{3.i.2}$ except that for the first j times it is queried, it outputs $\perp$ to any ill-formed query. EncO is as in $\mathrm{G}_{3.i.2}$.

We show that:

$$\begin{aligned}
&\mathrm{G}_{3.i.1} \equiv \mathrm{G}_{3.i.1.0} \approx_{\mathsf{AE}} \mathrm{G}_{3.i.1.1} \approx_{\mathsf{AE}} \ldots \approx_{\mathsf{AE}} \mathrm{G}_{3.i.1.Q_{\mathsf{dec}}} \equiv \mathrm{G}'_{3.i.1.Q_{\mathsf{dec}}} \\
&\mathrm{G}'_{3.i.1.Q_{\mathsf{dec}}} \approx_{\mathsf{AE}} \mathrm{G}'_{3.i.1.Q_{\mathsf{dec}}-1} \approx_{\mathsf{AE}} \ldots \approx_{\mathsf{AE}} \mathrm{G}'_{3.i.1.0} \equiv \mathrm{G}_{3.i.2}
\end{aligned}$$

where $\equiv$ denote statistical equality, and $\approx_{\mathsf{AE}}$ denotes indistinguishability based on the security of AE.

It suffices to show that for all $j = 0, \ldots, Q_{\mathsf{dec}}-1$, there exist adversaries $\mathcal{B}_{3.i.1.j}$ and $\mathcal{B}'_{3.i.1.j}$ against the one-time authenticity of AE, such that $\mathbf{T}(\mathcal{B}_{3.i.1.j}) \approx \mathbf{T}(\mathcal{B}'_{3.i.1.j}) \approx \mathbf{T}(\mathcal{A}) + (Q_{\mathsf{enc}} + Q_{\mathsf{dec}}) \cdot \mathsf{poly}(\lambda)$, with $\mathsf{poly}(\lambda)$ independent of $\mathbf{T}(\mathcal{A})$, and such that:

Claim 1: in $\mathrm{G}_{3.i.1.j}$, if the $j+1$-st query is ill-formed, then DecO outputs $\perp$ with overwhelming probability $1 - \mathbf{Adv}^{\mathsf{ae\text{-}ot}}_{\mathsf{AE}}(\mathcal{B}_{3.i.1.j})$ (this implies $\mathrm{G}_{3.i.1.j} \approx_{\mathsf{AE}} \mathrm{G}_{3.i.1.j+1}$).

Claim 2: in $\mathrm{G}'_{3.i.1.j}$, if the $j+1$-st query is ill-formed, then DecO outputs 0 with overwhelming probability $1 - \mathbf{Adv}^{\mathsf{ae\text{-}ot}}_{\mathsf{AE}}(\mathcal{B}'_{3.i.1.j})$ (this implies $\mathrm{G}'_{3.i.1.j} \approx_{\mathsf{AE}} \mathrm{G}'_{3.i.1.j+1}$),

where the probabilities are taken over the random coins of Setup.

We prove Claim 1 and 2 as in Lemma 8, in Section 3, arguing that the encapsulation key K computed by DecO on an ill-formed $j+1$-st query, is completely hidden from $\mathcal{A}$, up to its $j+1$-st query to DecO. The reason is that the vector $\mathbf{k}_{i+1,1}$ in sk contains some entropy that is hidden from $\mathcal{A}$, and that is "released" on the $j+1$-st query, if it is ill-formed. Then, we use the one-time authenticity of AE to argue that DecO outputs $\perp$ with overwhelming probability over the random coins of Setup. □

Lemma 18 ($\mathrm{G}_{3.i.2}$ to $\mathrm{G}_{3.i.3}$). *For all $0 \leq i \leq \lambda - 1$, there exists an adversary $\mathcal{B}_{3.i.2}$ such that $\mathbf{T}(\mathcal{B}_{3.i.2}) \approx \mathbf{T}(\mathcal{A}) + (Q_{\mathsf{enc}} + Q_{\mathsf{dec}}) \cdot \mathsf{poly}(\lambda)$,*

$$|\mathbf{Adv}_{3.i.2} - \mathbf{Adv}_{3.i.3}| \leq 2Q_{\mathsf{dec}} \cdot \mathbf{Adv}^{\mathsf{ae\text{-}ot}}_{\mathsf{AE}}(\mathcal{B}_{3.i.2})$$

where Q_{enc}, Q_{dec} are the number of times $\mathcal{A}$ queries DecO, and $\mathsf{poly}(\lambda)$ is independent of $\mathbf{T}(\mathcal{A})$.

Proof of Lemma 18. In $\mathrm{G}_{3.i.3}$, we use the same decomposition $\mathsf{span}(\mathbf{M}^{\perp}) = \mathsf{span}(\mathbf{M}^*_0, \mathbf{M}^*_1)$ as that in $\mathrm{G}_{3.i.2}$. The entropy of the component of $\mathbf{k}'_\tau$ that lies in $\mathsf{span}(\mathbf{M}^*_1)$ increases from $\mathrm{G}_{3.i.2}$ to $\mathrm{G}_{3.i.3}$. That is, we use a random function $\mathsf{RF}^{(1)}_{i+1} : \{0,1\}^{i+1} \to \mathbb{Z}^k_q$ in place of the random function $\mathsf{RF}^{(1)}_i : \{0,1\}^i \to \mathbb{Z}^k_q$. To argue that these two games are computationally indistinguishable, we use a computational variant of the Cramer-Shoup argument [11], exactly as in the proof of Lemma 17.

We define $\mathsf{RF}^{(1)}_{i+1} \to \mathbb{Z}^k_q$ as follows:

$$\mathsf{RF}^{(1)}_{i+1}(\tau_{|i+1}) := \begin{cases} \mathsf{RF}^{(1)}_i(\tau_{|i}) + \mathsf{RF}'^{(1)}_i(\tau_{|i}) & \text{if } \tau_{i+1} = 0 \\ \mathsf{RF}^{(1)}_i(\tau_{|i}) & \text{if } \tau_{i+1} = 1 \end{cases}$$

where $\mathsf{RF}'^{(1)}_i : \{0,1\}^i \to \mathbb{Z}^k_q$ is a random function independent from $\mathsf{RF}^{(1)}_i$. This way, $\mathsf{RF}^{(1)}_{i+1}$ is a random function.

We show that the outputs of EncO and DecO are computationally indistinguishable in $\mathrm{G}_{3.i.1}$ and $\mathrm{G}_{3.i.2}$, similarly that in the proof of Lemma 9, in Section 3 (see the latter for further details). □

Lemma 19 ($\mathrm{G}_{3.i.3}$ to $\mathrm{G}_{3.i+1}$). *For all $0 \leq i \leq \lambda - 1$, there exists an adversary $\mathcal{B}_{3.i.3}$ such that $\mathbf{T}(\mathcal{B}_{3.i.3}) \approx \mathbf{T}(\mathcal{A}) + (Q_{\mathsf{enc}} + Q_{\mathsf{dec}}) \cdot \mathsf{poly}(\lambda)$ and*

$$|\mathbf{Adv}_{3.i.3} - \mathbf{Adv}_{3.i+1}| \leq 2 \cdot \mathbf{Adv}^{\mathrm{mddh}}_{\mathcal{U}_k,\mathsf{GGen}}(\mathcal{B}_{3.i.3}) + \frac{2}{q-1},$$

where Q_{enc}, Q_{dec} are the number of times $\mathcal{A}$ queries EncO, DecO, respectively, and $\mathsf{poly}(\lambda)$ is independent of $\mathbf{T}(\mathcal{A})$.

Here, we use the MDDH Assumption to "tightly" switch the distribution of all the challenge ciphertexts. As for Lemma 16, we proceed in two steps, and we use an independent matrix from $\mathcal{U}_{2k,k}$ for each step.

Proof of Lemma 19. To go from $\mathrm{G}_{3.i.3}$ to $\mathrm{G}_{3.i+1}$, we switch the distribution of the vector $[\mathbf{y}]$ sampled by EncO, using the Q_{enc}-fold $\mathcal{U}_{2k,k}$-MDDH Assumption (equivalent to the $\mathcal{U}_k$-MDDH Assumption, see

Lemma 1). This transition is symmetric to the transition between $\mathrm{G}_{3.i}$ and $\mathrm{G}_{3.i.1}$, and we defer to the proof of Lemma 16 for further details. Finally, we use the fact that for all $\tau \in \{0,1\}^\lambda$, $\mathbf{M}_0^* \mathsf{RF}_{i+1}^{(0)}(\tau_{|i}) + \mathbf{M}_1^* \mathsf{RF}_{i+1}^{(1)}(\tau_{|i+1})$ is identically distributed to $\mathbf{M}^\perp \mathsf{RF}_{i+1}(\tau_{|i+1})$, where $\mathsf{RF}_{i+1} : \{0,1\}^{i+1} \to \mathbb{Z}_q^{2k}$ is a random function. This is because $(\mathbf{M}_0^*, \mathbf{M}_1^*)$ is a basis of $\mathsf{span}(\mathbf{M}^\perp)$. □

The proof of Lemma 15 follows readily from Lemmas 16, 17, 18, and 19. □

Lemma 20 ($\mathrm{G}_{3.\lambda}$ to G_4). *There exists an adversary $\mathcal{B}_{3.\lambda}$ such that $\mathbf{T}(\mathcal{B}_{3.\lambda}) \approx \mathbf{T}(\mathcal{A}) + (Q_{\mathsf{enc}} + Q_{\mathsf{dec}}) \cdot \mathsf{poly}(\lambda)$, and*

$$|\mathbf{Adv}_{3.\lambda} - \mathbf{Adv}_4| \leq Q_{\mathsf{dec}} Q_{\mathsf{enc}} \cdot \mathbf{Adv}^{\mathsf{ae\text{-}ot}}_{\mathsf{AE}}(\mathcal{B}_{3.\lambda}) + \frac{Q_{\mathsf{dec}}}{q},$$

where Q_{enc}, Q_{dec} are the number of times $\mathcal{A}$ queries DecO, and $\mathsf{poly}(\lambda)$ is independent of $\mathbf{T}(\mathcal{A})$.

Here, we use the one-time authenticity of AE to restrict the decryption oracle DecO.

Proof of Lemma 20. We use the one-time authenticity of AE to argue that with overwhelming probability over the random coins of Setup, DecO outputs $\perp$ on any input $([\mathbf{y}], \phi)$ such that for some previous output $([\mathbf{y}'], \phi')$ of EncO, $\mathsf{H}([\overline{\mathbf{y}'}]) = \mathsf{H}([\overline{\mathbf{y}}])$.

We introduce intermediate games $\mathrm{G}_{3.\lambda.j}$ for $j = 0, \ldots, Q_{\mathsf{dec}}$, defined as $\mathrm{G}_{3.\lambda}$, except that on its first j query, DecO is as in G_4, that is, it outputs $\perp$ to any query corresponding to a tag τ previously output by EncO.

We show that :

$$\mathrm{G}_{3.\lambda} \equiv \mathrm{G}_{3.\lambda.0} \approx_{\mathsf{AE}} \mathrm{G}_{3.\lambda.1} \approx_{\mathsf{AE}} \ldots \approx_{\mathsf{AE}} \mathrm{G}_{3.\lambda.Q_{\mathsf{dec}}} \equiv \mathrm{G}_4,$$

where $\equiv$ denotes statistical equality, and $\approx_{\mathsf{AE}}$ denotes indistinguishability based on the security of AE.

Namely, we build adversaries $\mathcal{B}_{3.\lambda.j}$ for $j = 0, \ldots, Q_{\mathsf{dec}} - 1$, such that $\mathbf{T}(\mathcal{B}_{3.\lambda.j}) \approx \mathbf{T}(\mathcal{A}) + (Q_{\mathsf{enc}} + Q_{\mathsf{dec}}) \cdot \mathsf{poly}(\lambda)$, where $\mathsf{poly}(\lambda)$ is independent of $\mathbf{T}(\mathcal{A})$, and

$$|\mathbf{Adv}_{3.\lambda.j} - \mathbf{Adv}_{3.\lambda.j+1}| \leq Q_{\mathsf{enc}} \cdot \mathbf{Adv}^{\mathsf{ae\text{-}ot}}_{\mathsf{AE}}(\mathcal{B}_{3.\lambda.j}) + \frac{1}{q}.$$

This implies the lemma.

It suffices to show that in $\mathrm{G}_{3.\lambda.j}$, with overwhelming probability over the random coins of $\mathcal{A}$, Setup and EncO: DecO outputs $\perp$ to its $j+1$-st query if it contains $[\mathbf{y}^\star]$ such that $\mathsf{H}([\overline{\mathbf{y}^\star}]) = \mathsf{H}([\overline{\mathbf{y}}])$, for $[\mathbf{y}]$ that was output previously by EncO.

We build $\mathcal{B}_{3.\lambda.j}$ as follows.

Setup **:** Upon receiving the description of $\mathcal{K} := \mathcal{G}$, $\mathcal{B}_{3.\lambda.j}$ picks $\mathbf{M} \leftarrow_{\mathrm{R}} \mathcal{U}_{3k,k}$, $\mathbf{k}_{1,0}, \ldots, \mathbf{k}_{\lambda,1} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k}$, $\mathsf{H} \leftarrow_{\mathrm{R}} \mathcal{H}(1^\lambda)$, and outputs pk as in G_4 (see Figure 9). It also picks $j^\star \leftarrow_{\mathrm{R}} \{1, \ldots, Q_{\mathsf{enc}}\}$, and $b \leftarrow_{\mathrm{R}} \{0,1\}$.

$\mathsf{EncO}(M_0, M_1)$ **:** On the $j^\star$'th query, $\mathcal{B}_{3.\lambda.j}$ picks $\mathbf{y} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k}$, calls $\mathsf{ot\text{-}EncO}(M_b, M_b)$ to get $\phi_b := \mathsf{Enc}_{\mathsf{AE}}(K^\star, M_b)$, for a random $K^\star \leftarrow_{\mathrm{R}} \mathbb{G}$. The rest of the simulation goes as in G_4 (see Figure 9), that is: if $\mathsf{H}([\overline{\mathbf{y}}]) \notin \mathcal{T}_{\mathsf{enc}} \cup \mathcal{T}_{\mathsf{dec}}$, $\mathcal{B}_{3.\lambda.j}$ returns $([\mathbf{y},], \phi_b)$, sets $\mathcal{T}_{\mathsf{enc}} := \mathcal{T}_{\mathsf{enc}} \cup \{\mathsf{H}([\overline{\mathbf{y}}])\}$ and $\mathcal{C}_{\mathsf{enc}} := \mathcal{C}_{\mathsf{enc}} \cup \{([\mathbf{y}], \phi_b)\}$, otherwise, it returns $\perp$. The other $j \neq j^\star$ queries are simulated as in G_4.

$\mathsf{DecO}([\mathbf{y}], \phi)$**:** the first j queries are simulated as in G_4, the last $Q_{\mathsf{enc}} - j - 1$ as in $\mathrm{G}_{3.\lambda}$. For the $j+1$-st query $([\mathbf{y}^\star], \phi^\star)$, $\mathcal{B}_{3.\lambda.j}$ calls $\mathsf{ot\text{-}DecO}([\mathbf{y}^\star], \phi^\star)$ to get $\mathsf{Dec}_{\mathsf{AE}}(K^\star, \phi^\star)$. The rest of the simulation goes as in $\mathrm{G}_{3.i}$, that is, if $([\mathbf{y}^\star], \phi^\star) \in \mathcal{C}_{\mathsf{enc}}$ or $\exists([\mathbf{y}], \phi) \in \mathcal{C}_{\mathsf{enc}}$ with $\mathsf{H}([\overline{\mathbf{y}}^\star]) = \mathsf{H}([\overline{\mathbf{y}}])$ and $\mathbf{y}^\star \neq \mathbf{y}$, $\mathcal{B}_{3.\lambda.j}$ returns $\perp$. Otherwise, it returns $\mathsf{Dec}_{\mathsf{AE}}(K^\star, \phi^\star)$. Finally, it sets $\mathcal{T}_{\mathsf{dec}} := \mathcal{T}_{\mathsf{dec}} \cup \{\mathsf{H}([\overline{\mathbf{y}}^\star])\}$.

Assume the $j+1$-st query $([\mathbf{y}^\star], \phi^\star)$ to DecO is such that $\mathsf{DecO}([\mathbf{y}^\star], \phi^\star) = \perp$ in G_4, but not in $\mathrm{G}_{3.\lambda.j}$. In particular, that means that there exists $([\mathbf{y}], \phi) \in \mathcal{C}_{\mathsf{enc}}$ such that $\mathbf{y} = \mathbf{y}^\star$ and $\phi \neq \phi^\star$. Then, with probability $1/Q_{\mathsf{enc}}$ over the choice of $j^\star$, $([\mathbf{y}], \phi)$ is the $j^\star$'th query of EncO. In that case, we show that $\mathcal{A}$'s view is simulated as in $\mathrm{G}_{3.\lambda.j}$ if $\mathsf{ot\text{-}DecO}$ is the real decryption oracle, and as in G_4 if it is the "always $\perp$" function. This implies the lemma.

Indeed, the key $K^\star := [\mathbf{y}^{\star\top}(\mathbf{k}_{\tau^\star} + \mathbf{M}^\perp \mathsf{RF}_\lambda(\tau^\star))]$ for $\tau^\star := \mathsf{H}([\overline{\mathbf{y}}^\star])$ is random, independent from $\mathcal{A}$'s view up to its $j+1$-st query on DecO (except what leaks through $\mathsf{Enc}_{\mathsf{AE}}(K^\star, M_b)$). This is because:

1. with probability $1/q$ over the random coins of $\mathcal{B}_{3.\lambda.j}$, $\mathbf{y}^\star \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k} \notin \mathsf{span}(\mathbf{M})$.
2. for all $[\mathbf{y}]$ contained in EncO outputs or DecO queries that don't output $\perp$, prior to the $j+1$-st DecO query, we have $\mathsf{H}([\overline{\mathbf{y}}]) \neq \tau^\star$, by definition of $\mathrm{G}_{3.\lambda.j}$. That is, the tag $\tau^\star$ is "fresh". Therefore, the key
$$K^\star := [\mathbf{y}^{\star\top}(\mathbf{k}_{\tau^\star} + \mathbf{M}^\perp \mathsf{RF}_\lambda(\tau^\star))] = [\mathbf{y}^\top \mathbf{k}_{\tau^\star} + \underbrace{\mathbf{y}^{\star\top}\mathbf{M}^\perp}_{\neq \mathbf{0}} \mathsf{RF}_\lambda(\tau^\star)]$$
is random, independent of $\mathcal{A}$'s view up to its $j+1$-st query (except what leaks through $\mathsf{Enc}_{\mathsf{AE}}(K^\star, M_b)$).

This proves that
$$|\mathbf{Adv}_{3.\lambda.j} - \mathbf{Adv}_{3.\lambda.j+1}| \leq Q_{\mathsf{enc}} \cdot \mathbf{Adv}^{\mathsf{ae\text{-}ot}}_{\mathsf{AE}}(\mathcal{B}_{3.\lambda.j}) + \frac{1}{q}.$$

□

Lemma 21 (G_4)**.** *There exists an adversary $\mathcal{B}_4$ such that $\mathbf{T}(\mathcal{B}_4) \approx \mathbf{T}(\mathcal{A}) + (Q_{\mathsf{enc}} + Q_{\mathsf{dec}}) \cdot \mathsf{poly}(\lambda)$, such that*
$$\mathbf{Adv}_4 \leq Q_{\mathsf{enc}} \cdot \mathbf{Adv}^{\mathsf{ae\text{-}ot}}_{\mathsf{AE}}(\mathcal{B}_4) + \frac{Q_{\mathsf{enc}}}{q},$$
where Q_{enc} and Q_{dec} are the number of times $\mathcal{A}$ queries DecO, and $\mathsf{poly}(\lambda)$ is independent of $\mathbf{T}(\mathcal{A})$.

Proof of Lemma 21. First, we show that the joint distribution of all the values K computed by EncO is statistically close to uniform over $\mathbb{G}^{Q_{\mathsf{enc}}}$. Then, we use the one-time privacy of AE on each one of the Q_{enc} symmetric ciphertexts.

Recall that on input τ, $\mathsf{EncO}(\tau)$ computes
$$K := [\mathbf{y}^\top(\mathbf{k}_\tau + \mathbf{M}^\perp \mathsf{RF}_\lambda(\tau))],$$
where $\mathsf{RF}_\lambda : \{0,1\}^\lambda \to \mathbb{Z}_q^{2k}$ is a random function, and $\mathbf{y} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k}$.

We make use of the following properties:

Property 1: all the tags τ computed by $\mathsf{EncO}(M_0, M_1)$, such that $\mathsf{EncO}(M_0, M_1) \neq \perp$, are distinct.
Property 2: the outputs of DecO are independent of $\{\mathsf{RF}(\tau) : \tau \in \mathcal{T}_{\mathsf{enc}}\}$. This is because for all queries $([\mathbf{y}], \phi)$ to DecO such that $\mathsf{H}([\overline{\mathbf{y}}]) \in \mathcal{T}_{\mathsf{enc}}$, $\mathsf{DecO}([\mathbf{y}], \phi) = \perp$, independently of $\mathsf{RF}_\lambda(\tau)$, by definition of G_4.

Property 3: with probability at least $1 - \frac{Q_{\mathsf{enc}}}{q}$ over the random coins of EncO, all the vectors $\mathbf{y}$ sampled by EncO are such that $\mathbf{y}^\top \mathbf{M}^\perp \neq \mathbf{0}$.

We deduce that the joint distribution of all the values $\mathsf{RF}_\lambda(\tau)$ computed by EncO is uniformly random over $\left(\mathbb{Z}_q^{2k}\right)^{Q_{\mathsf{enc}}}$ (from Property 1), independent of the outputs of DecO (from Property 2). Finally, from Property 3, we get that the joint distribution of all the values K computed by EncO is statistically close to uniformly random over $\mathbb{G}^{Q_{\mathsf{enc}}}$, since:

$$K := [\mathbf{y}^\top(\mathbf{k}_\tau + \mathbf{M}^\perp \mathsf{RF}_\lambda(\tau)) = [\mathbf{y}^\top \mathbf{k}_\tau + \underbrace{\mathbf{y}^\top \mathbf{M}^\perp}_{\neq \mathbf{0} \text{ w.h.p.}} \mathsf{RF}_\lambda(\tau)].$$

Therefore, we can use the one-time privacy of AE to argue that all symmetric ciphertexts ϕ_b computed by EncO don't reveal b (this uses a Q_{enc}-hybrid argument). □

Finally, Theorem 3 follows readily from Lemmas 12, 13,14, 15, 20, and 21. □

6 Tightly secure, Quasi-adaptive Zero-Knowledge arguments for Linear Subspaces

Here, we show how we can apply our PCA-secure KEM of Section 3 to obtain tightly secure, (Designated-Verifier) Quasi-Adaptive Non-Interactive Zero-Knowledge arguments ((DV)QANIZK) for linear subspaces, with strong simulation soundness. In Section 6.1, we recall the definitions of QANIZK and DVQANIZK arguments. In Section 6.2, we give a generic construction of a DVQANIZK argument for linear language, from a PCA-secure KEM and a concrete instantiation of this generic construction, using the PCA-secure KEM presented in Section 3. Finally, in Section 6.3, we give a QANIZK argument for linear language, which is more efficient than simply upgrading the DVQANIZK in Section 6.2 with pairings.

6.1 Quasi-adaptive Non-Interactive Zero-Knowledge

Quasi-Adaptive NIZK (QA-NIZK) proofs are NIZK proofs where the common reference string (CRS) is allowed to depend on the specific language for which proofs have to be generated [20]. The CRS is generated in a specific way and contains a fixed part par, produced by an algorithm $\mathsf{Gen}_{\mathsf{par}}$, and a language-dependent part crs. However, for the zero-knowledge property there should exist a single simulator for the entire class of languages.

For public parameters par produced by $\mathsf{Gen}_{\mathsf{par}}$, let $\mathcal{D}_{\mathsf{par}}$ be a probability distribution over a collection of relations $R = \{R_\rho\}$ parametrized by a string ρ with an associated language $\mathcal{L}_\rho = \{y : \exists x \text{ s.t. } R_\rho(y, x) = 1\}$.

We now give a formal definition of QANIZK for $\mathcal{D}_{\mathsf{par}}$ in its tag-based variant. The tag-based version can be transformed into a standard QANIZK using a one-time signature.

Definition 8 (QANIZK Argument). *A* Quasi-adaptive Non-Interactive Zero Knowledge Argument *(QANIZK) Π for a language distribution $\mathcal{D}_{\mathsf{par}}$ consists of five PPT algorithms $\Pi = (\mathsf{Gen}_{\mathsf{par}}, \mathsf{Gen}_{\mathsf{crs}}, \mathsf{Prove}, \mathsf{Sim}, \mathsf{Ver})$:*

- *The probabilistic key generation algorithm $\mathsf{Gen}_{\mathsf{par}}(1^\lambda)$ returns the public parameters par.*

– *The probabilistic algorithm* $\mathsf{Gen}_{\mathsf{crs}}(\mathsf{par}, \rho)$ *returns a common reference string* crs*, and a trapdoor* trap*. We assume that* crs *implicitly contains* par *and* ρ*, and that it defines a tag-space* $\mathcal{T}$*. (This is the classical QANIZK setting.) If* $\mathcal{T}$ *is not specified then* $\mathcal{T} = \{\varepsilon\}$ *and tags can be ignored in all algorithms.*
– *The probabilistic proving algorithm* $\mathsf{Prove}(\mathsf{crs}, \tau, x, y)$ *returns a proof* π*, with respect to tag* $\tau \in \mathcal{T}$*.*
– *The probabilistic verification algorithm* $\mathsf{Ver}(\mathsf{crs}, \tau, y, \pi)$ *returns 1 or* 0*, where* 1 *means that* π *is a valid proof of* $y \in \mathcal{L}_\rho$*.*
– *The probabilistic proving algorithm* $\mathsf{Sim}(\mathsf{crs}, \mathsf{trap}, \tau, y)$ *returns a proof* π *for some* $y \in \mathcal{Y}$ *(not necessarily in* $\mathcal{L}_\rho$*) with respect to tag* $\tau \in \mathcal{T}$*.*

We require that the algorithms satisfy the following properties:

Perfect completeness. *For all* λ*, all* par *output by* $\mathsf{Gen}_{\mathsf{par}}(\lambda)$*, all* ρ *output by* $\mathcal{D}_{\mathsf{par}}$*, all* (x, y) *with* $R_\rho(y, x) = 1$*, all* $\tau \in \mathcal{T}$*, we have*

$$\Pr[\mathsf{Ver}(\mathsf{crs}, \tau, y, \pi) = 1 | (\mathsf{crs}, \mathsf{trap}) \leftarrow_{\mathrm{R}} \mathsf{Gen}_{\mathsf{crs}}(\mathsf{par}, \rho); \pi \leftarrow_{\mathrm{R}} \mathsf{Prove}(\mathsf{crs}, \tau, x, y)] = 1.$$

Perfect zero-knowledge. *For all* λ*, all* par *output by* $\mathsf{Gen}_{\mathsf{par}}(\lambda)$*, all* ρ *output by* $\mathcal{D}_{\mathsf{par}}$*, all* $(\mathsf{crs}, \mathsf{trap})$ *output by* $\mathsf{Gen}_{\mathsf{crs}}(\mathsf{par}, \rho)$*, all* (x, y) *with* $R_\rho(y, x) = 1$*, all* $\tau \in \mathcal{T}$*, the distributions*

$$\mathsf{Prove}(\mathsf{crs}, \tau, x, y) \text{ and } \mathsf{Sim}(\mathsf{crs}, \mathsf{trap}, \tau, y)$$

are the same (where the coin tosses are taken over $\mathsf{Prove}, \mathsf{Sim}$*).*

Unbounded Simulation Soundness [33, 12]. *For all PPT adversaries* $\mathcal{A}$*,*

$$\mathbf{Adv}^{\mathrm{uss}}_{\Pi}(\mathcal{A}) := \Pr\left[\mathrm{WIN} = 1 | \mathcal{A}^{\mathsf{Setup}, \mathsf{SimO}(\cdot,\cdot), \mathsf{VerO}(\cdot,\cdot,\cdot)}\right]$$

is negligible, where:

– Setup *sets* $\mathrm{WIN} := 0$*,* $\mathcal{T}_{\mathsf{sim}} := \emptyset$*, then samples* $\mathsf{par} \leftarrow_{\mathrm{R}} \mathsf{Gen}_{\mathsf{par}}(\lambda); \rho \leftarrow_{\mathrm{R}} \mathcal{D}_{\mathsf{par}}$*;* $(\mathsf{crs}, \mathsf{trap}) \leftarrow_{\mathrm{R}} \mathsf{Gen}_{\mathsf{crs}}(\mathsf{par}, \rho)$ *and returns* crs*. We require that* Setup *must be called once at the beginning of the game.*
– $\mathsf{SimO}(\tau, y)$ *returns* $\pi := \mathsf{Sim}(\mathsf{crs}, \mathsf{trap}, \tau, y)$ *and sets* $\mathcal{T}_{\mathsf{sim}} := \mathcal{T}_{\mathsf{sim}} \cup \{\tau\}$*.*
– $\mathsf{VerO}(\tau, y, \pi)$ *sets* $\mathrm{WIN} = 1$ *if* $\mathsf{Ver}(\mathsf{crs}, \tau, y, \pi) = 1 \wedge y \notin \mathcal{L}_\rho \wedge \tau \notin \mathcal{T}_{\mathsf{sim}}$*.* VerO *is called at most once.*

Now, we give the definition of Designated-Verifier QANIZK (DVNIZK) arguments in their tag based variant. Roughly speaking, a DVQANIZK is a QANIZK where a secret verification key vk is needed to verify the membership of an instance, unlike a regular QANIZK where only the crs is needed.

Definition 9 (DVQANIZK Argument). *A* Designated-Verifier, Quasi-adaptive Non-Interactive Zero Knowledge Argument *(DVQANIZK)* Π *for a language distribution* $\mathcal{D}_{\mathsf{par}}$ *consists of five PPT algorithms* $\Pi = (\mathsf{Gen}_{\mathsf{par}}, \mathsf{Gen}_{\mathsf{crs}}, \mathsf{Prove}, \mathsf{Sim}, \mathsf{Ver})$*:*

– *The probabilistic key generation algorithm* $\mathsf{Gen}_{\mathsf{par}}(1^\lambda)$ *returns the public parameters* par*.*
– *The probabilistic algorithm* $\mathsf{Gen}_{\mathsf{crs}}(\mathsf{par}, \rho)$ *returns a common reference string* crs*, a trapdoor* trap*, and a verification key* vk*. We assume that* crs *implicitly contains* par *and* ρ*, and that it defines a tag-space* $\mathcal{T}$*. If* $\mathcal{T}$ *is not specified then* $\mathcal{T} = \{\varepsilon\}$ *and tags can be ignored in all algorithms.*
– *The probabilistic proving algorithm* $\mathsf{Prove}(\mathsf{crs}, \tau, x, y)$ *returns a proof* π*, with respect to tag* $\tau \in \mathcal{T}$*.*

– *The probabilistic verification algorithm* $\mathsf{Ver}(\mathsf{crs}, \boxed{\mathsf{vk}}, \tau, y, \pi)$ *returns 1 or* 0*, where* 1 *means that* π *is a valid proof of* $y \in \mathcal{L}_\rho$*.*
– *The probabilistic proving algorithm* $\mathsf{Sim}(\mathsf{crs}, \mathsf{trap}, \tau, y)$ *returns a proof* π *for some* $y \in \mathcal{Y}$ *(not necessarily in* $\mathcal{L}_\rho$*) with respect to tag* $\tau \in \mathcal{T}$*.*

We require that the algorithms satisfy the following properties:

Perfect completeness. *For all* λ*, all* par *output by* $\mathsf{Gen}_{\mathsf{par}}(\lambda)$*, all* ρ *output by* $\mathcal{D}_{\mathsf{par}}$*, all* (x, y) *with* $R_\rho(y, x) = 1$*, all* $\tau \in \mathcal{T}$*, we have*

$$\Pr[\mathsf{Ver}(\mathsf{crs}, \mathsf{vk}, \tau, y, \pi) = 1 | (\mathsf{crs}, \mathsf{trap}, \mathsf{vk}) \leftarrow_{\mathrm{R}} \mathsf{Gen}_{\mathsf{crs}}(\mathsf{par}, \rho); \pi \leftarrow_{\mathrm{R}} \mathsf{Prove}(\mathsf{crs}, \tau, x, y)] = 1.$$

Perfect zero-knowledge. *For all* λ*, all* par *output by* $\mathsf{Gen}_{\mathsf{par}}(\lambda)$*, all* ρ *output by* $\mathcal{D}_{\mathsf{par}}$*, all* $(\mathsf{crs}, \mathsf{trap})$ *output by* $\mathsf{Gen}_{\mathsf{crs}}(\mathsf{par}, \rho)$*, all* (x, y) *with* $R_\rho(y, x) = 1$*, all* $\tau \in \mathcal{T}$*, the distributions*

$$\mathsf{Prove}(\mathsf{crs}, \tau, x, y) \text{ and } \mathsf{Sim}(\mathsf{crs}, \mathsf{trap}, \tau, y)$$

are the same (where the coin tosses are taken over $\mathsf{Prove}, \mathsf{Sim}$*).*

Strong Unbounded Simulation Soundness. *For all PPT adversaries* $\mathcal{A}$*,*

$$\mathbf{Adv}^{\mathrm{uss}}_{\Pi}(\mathcal{A}) := \Pr\left[\begin{matrix}\exists(\tau^\star, y^\star, \pi^\star) \in \mathcal{Q}_{\mathrm{ver}} \text{ s.t.} \\ y^\star \notin \mathcal{L}_\rho \wedge \mathsf{VerO}(\tau^\star, y^\star, \pi^\star) = 1\end{matrix} \middle| \mathcal{A}^{\mathsf{Setup}, \mathsf{SimO}(\cdot,\cdot), \mathsf{VerO}(\cdot,\cdot,\cdot)}\right]$$

is negligible, where:
– Setup *sets* $\mathcal{Q}_{\mathrm{ver}} = \mathcal{T}_{\mathsf{sim}} = \mathcal{T}_{\mathsf{ver}} := \emptyset$*, samples* $\mathsf{par} \leftarrow_{\mathrm{R}} \mathsf{Gen}_{\mathsf{par}}(\lambda); \rho \leftarrow_{\mathrm{R}} \mathcal{D}_{\mathsf{par}}$*;* $(\mathsf{crs}, \mathsf{trap}, \mathsf{vk}) \leftarrow_{\mathrm{R}} \mathsf{Gen}_{\mathsf{crs}}(\mathsf{par}, \rho)$ *and returns* crs*.* Setup *is called once at the beginning of the game.*
– $\mathsf{SimO}(\tau, y)$*: if* $\tau \notin \mathcal{T}_{\mathsf{ver}} \cup \mathcal{T}_{\mathsf{sim}}$*, it returns* $\pi := \mathsf{Sim}(\mathsf{crs}, \mathsf{trap}, \tau, y)$ *and sets* $\mathcal{T}_{\mathsf{sim}} := \mathcal{T}_{\mathsf{sim}} \cup \{\tau\}$*; otherwise, it returns* $\perp$*.*
– $\mathsf{VerO}(\tau, y, \pi)$ *returns* 1 *if* $\mathsf{Ver}(\mathsf{crs}, \mathsf{vk}, \tau, y, \pi) = 1 \wedge \tau \notin \mathcal{T}_{\mathsf{sim}}$*,* 0 *otherwise. Then it sets* $\mathcal{Q}_{\mathrm{ver}} := \mathcal{Q}_{\mathrm{ver}} \cup \{(\tau, y, \pi)\}$*,* $\mathcal{T}_{\mathsf{ver}} := \mathcal{T}_{\mathsf{ver}} \cup \{\tau\}$*.*

Unbounded Simulation Soundness. *This property is defined as Strong Unbounded Simulation Soundness except the adversary is only allowed one call to* VerO*. This is the standard notion of unbounded simulation soundness [33, 12].*

6.2 Generic construction of DVQANIZK argument for linear subspace, with strong simulation soundness

In this section we describe a tightly-secure, Designated-Verifier Quasi-adaptive Non-Interactive Zero Knowledge Argument for linear subspaces with strong unbounded simulation-soundness (see Definition 9).

We use $\mathsf{Gen}_{\mathsf{par}} = \mathsf{GGen}$. That is, $\mathsf{Gen}_{\mathsf{par}}(1^\lambda)$ returns $\mathsf{par} = \mathcal{G}$, where $\mathcal{G} = (\mathbb{G}, q, g)$ contains a cyclic group $\mathbb{G}$ generated by g of order q (see Section 2.3). The probability distribution $\mathcal{D}_{\mathsf{par}}$ returns a matrix $\rho = [\mathbf{M}] \in \mathbb{G}^{n \times t}$, for integers $n > t$. We consider the case of *witness sampleable* (WS) [20] distributions, where there exist an efficiently sampleable distribution $\mathcal{D}'_{\mathsf{par}}$ that outputs $\mathbf{M}' \in \mathbb{Z}_q^{n \times t}$ such that $[\mathbf{M}']$ has the same distribution as $[\mathbf{M}]$. Note that this slightly restricts the set of languages that can be handled. Given par and ρ, the language $\mathcal{L}_{\mathbf{M}}$ is defined as

$$\mathcal{L}_{\mathbf{M}} = \left\{[\mathbf{y}] \in \mathbb{G}^n : \exists\, \mathbf{x} \in \mathbb{Z}_q^t \text{ s.t. } \mathbf{y} = \mathbf{M}\mathbf{x}\right\}.$$

The DVNIZK construction is given in Figure 11. When instantiated with the IND-PCA-secure KEM from Figure 4 (Section 3) we obtain the DVQANIZK argument described in Figure 12.

$\underline{\mathsf{Gen}_{\mathsf{crs}}(\mathsf{par}, [\mathbf{M}] \in \mathbb{G}^{n\times t})}$:
$(\mathsf{pk}, \mathsf{sk}) \leftarrow_{\mathrm{R}} \mathsf{Gen}_{\mathsf{KEM}}(1^\lambda)$
$\mathbf{k} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^n$
$\mathsf{crs} := \left([\mathbf{M}^\top \mathbf{k}], \mathsf{pk}\right)$
$\mathsf{trap} := \mathbf{k}$
$\mathsf{vk} := (\mathbf{k}, \mathsf{sk})$
Return $(\mathsf{crs}, \mathsf{trap}, \mathsf{vk})$.

$\underline{\mathsf{Prove}(\mathsf{crs}, \tau, [\mathbf{y}], \mathbf{x})}$: // $\mathbf{y} = \mathbf{M}\mathbf{x}$
$(C, K) \leftarrow_{\mathrm{R}} \mathsf{Enc}_{\mathsf{KEM}}(\mathsf{pk}, \tau)$
$[u] := [\mathbf{x}^\top \cdot \mathbf{M}^\top \mathbf{k}] + K$
Return $(C, [u])$

$\underline{\mathsf{Sim}(\mathsf{crs}, \mathsf{trap}, \tau, [\mathbf{y}])}$:
$(C, K) \leftarrow_{\mathrm{R}} \mathsf{Enc}_{\mathsf{KEM}}(\mathsf{pk}, \tau)$
$[u] := [\mathbf{y}^\top \cdot \mathbf{k}] + K$
Return $(C, [u])$

$\underline{\mathsf{Ver}(\mathsf{crs}, \mathsf{vk}, \tau, [\mathbf{y}], (C, [u]))}$:
$K := \mathsf{Dec}_{\mathsf{KEM}}(\mathsf{pk}, \mathsf{sk}, \tau, C)$
Return 1 if $K \neq \perp \wedge [u] = [\mathbf{y}^\top \cdot \mathbf{k}] + K$, 0 otherwise.

Fig. 11. DVQANIZK argument $\Pi_{\mathsf{uss}}^{\mathsf{dv}}$ with strong unbounded simulation-soundness, where $\mathsf{KEM}_{\mathsf{PCA}} := (\mathsf{Gen}_{\mathsf{KEM}}, \mathsf{Enc}_{\mathsf{KEM}}, \mathsf{Dec}_{\mathsf{KEM}})$ is an IND-PCA-secure KEM with key space $\mathcal{K} := \mathbb{G}$.

$\underline{\mathsf{Gen}_{\mathsf{crs}}(\mathcal{G}, [\mathbf{M}] \in \mathbb{G}^{n\times t})}$:
$\mathbf{B} \leftarrow_{\mathrm{R}} \mathcal{U}_{3k,k}$; $\mathbf{k} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^n$; $\mathbf{k}_{1,0}, \ldots, \mathbf{k}_{\lambda,1} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k}$
$\mathsf{crs} := \left([\mathbf{M}^\top \mathbf{k}], [\mathbf{B}], \left([\mathbf{B}^\top \mathbf{k}_{j,\beta}]\right)_{1\le j\le\lambda, 0\le\beta\le 1}\right)$
$\mathsf{trap} := \mathbf{k}$
$\mathsf{vk} := \left(\mathbf{k}, (\mathbf{k}_{j,\beta})_{1\le j\le\lambda, 0\le\beta\le 1}\right)$
Return $(\mathsf{crs}, \mathsf{trap}, \mathsf{vk})$.

$\underline{\mathsf{Prove}(\mathsf{crs}, \tau, [\mathbf{y}], \mathbf{x})}$: // $\mathbf{y} = \mathbf{M}\mathbf{x}$
$\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$; $\mathbf{k}_\tau := \sum_{j=1}^{\lambda} \mathbf{k}_{j,\tau_j}$
$\mathbf{t} := \mathbf{B}\mathbf{r}$
$[u] := [\mathbf{x}^\top \cdot \mathbf{M}^\top \mathbf{k} + \mathbf{r}^\top \cdot \mathbf{B}^\top \mathbf{k}_\tau]$
Return $([\mathbf{t}], [u]) \in \mathbb{G}^{3k+1}$

$\underline{\mathsf{Sim}(\mathsf{crs}, \mathsf{trap}, \tau, [\mathbf{y}])}$:
$\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$; $\mathbf{k}_\tau := \sum_{j=1}^{\lambda} \mathbf{k}_{j,\tau_j}$
$\mathbf{t} := \mathbf{B}\mathbf{r}$
$[u] := [\mathbf{y}^\top \cdot \mathbf{k} + \mathbf{r}^\top \cdot \mathbf{B}^\top \mathbf{k}_\tau]$
Return $([\mathbf{t}], [u]) \in \mathbb{G}^{3k+1}$

$\underline{\mathsf{Ver}(\mathsf{crs}, \mathsf{vk}, \tau, [\mathbf{y}], ([\mathbf{t}], [u]))}$:
$\mathbf{k}_\tau := \sum_{j=1}^{\lambda} \mathbf{k}_{j,\tau_j}$
Return 1 if $[u] = [\mathbf{y}^\top \cdot \mathbf{k} + \mathbf{t}^\top \mathbf{k}_\tau]$, 0 otherwise.

Fig. 12. DVQANIZK argument $\Pi_{\mathsf{uss}}^{\mathsf{dv}}$ with strong unbounded simulation-soundness under the $\mathcal{U}_k$-MDDH Assumption ($\Leftrightarrow \mathcal{U}_{3k,k}$-MDDH Assumption, by Lemma 1) and tag-space $\mathcal{T} = \{0,1\}^\lambda$.

Theorem 4. *The DVQANIZK argument $\Pi_{\mathsf{uss}}^{\mathsf{dv}}$ defined in Figure 11 has perfect zero-knowledge. Suppose in addition that the underlying KEM $\mathsf{KEM}_{\mathsf{PCA}}$ has perfect completeness, then, so does $\Pi_{\mathsf{uss}}^{\mathsf{dv}}$. Finally, if $\mathsf{KEM}_{\mathsf{PCA}}$ is* IND-PCA*-secure, then, $\Pi_{\mathsf{uss}}^{\mathsf{dv}}$ has strong unbounded simulation soundness. Namely, for any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ with $\mathbf{T}(\mathcal{A}) \approx \mathbf{T}(\mathcal{B}) + (Q_{\mathsf{sim}} + Q_{\mathsf{ver}}) \cdot \mathsf{poly}(\lambda)$ such that*

$$\mathbf{Adv}^{\mathrm{uss}}_{\Pi_{\mathsf{uss}}^{\mathsf{dv}}}(\mathcal{A}) \le 2\mathbf{Adv}^{\mathsf{ind\text{-}pca}}_{\mathsf{KEM}_{\mathsf{PCA}}}(\mathcal{B}) + \frac{Q_{\mathsf{ver}}}{q},$$

where Q_{sim}, Q_{ver}, is the number of times $\mathcal{A}$ queries SimO, VerO, *respectively, and $\mathsf{poly}(\lambda)$ is independent of $\mathbf{T}(\mathcal{A})$.*

Proof of Theorem 4. Perfect completeness and perfect zero-knowledge follow readily from the correctness of $\mathsf{KEM}_{\mathsf{PCA}}$, and the fact that for all $\mathbf{x} \in \mathbb{Z}_q^t$ and $\mathbf{y} = \mathbf{M}\mathbf{x}$, for all $\mathbf{k} \in \mathbb{Z}_q^n$:

$$\mathbf{x}^\top(\mathbf{M}^\top \mathbf{k}) = \mathbf{y}^\top \mathbf{k}.$$

We proceed to establish strong unbounded simulation soundness (see Definition 8), via a series of games described in Figure 13. We use $\mathbf{Adv}_i$ to denote the advantage of $\mathcal{A}$ in Game i.

Setup:	G_0,G_1	$\underline{\mathsf{SimO}([\mathbf{y}],\tau)}$:	G_0,$\boxed{G_1}$
$\mathcal{Q}_{\mathrm{ver}} = \mathcal{T}_{\mathsf{ver}} = \mathcal{T}_{\mathsf{sim}} := \emptyset$		$(K, C) \leftarrow_{\mathrm{R}} \mathsf{Enc}_{\mathsf{KEM}}(\mathsf{pk}, \tau)$	
$(\mathsf{pk}, \mathsf{sk}) \leftarrow_{\mathrm{R}} \mathsf{Gen}_{\mathsf{KEM}}(1^\lambda)$; $\mathbf{k} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^n$		$K' := K$; $\boxed{K' \leftarrow_{\mathrm{R}} \mathbb{G}}$	
$[\mathbf{M}] \leftarrow_{\mathrm{R}} \mathcal{D}_{\mathsf{par}}$		$[u] := [\mathbf{y}^\top \mathbf{k}] + K'$	
Return $\mathsf{crs} := ([\mathbf{M}^\top \mathbf{k}], \mathsf{pk})$.		If $\tau \notin \mathcal{T}_{\mathsf{sim}} \cup \mathcal{T}_{\mathsf{ver}}$, return $\pi := (C, [u])$ and set $\mathcal{T}_{\mathsf{sim}} := \mathcal{T}_{\mathsf{sim}} \cup \{\tau\}$; otherwise return $\perp$.	
//crs defines the tag-space $\mathcal{T} = \{0,1\}^\lambda$.			
		$\underline{\mathsf{VerO}(\tau, [\mathbf{y}], \pi := (C, [u]))}$:	G_0, G_1
		$K := \mathsf{Dec}_{\mathsf{KEM}}(\mathsf{sk}, \tau, C)$	
		Return 1 if $\tau \notin \mathcal{T}_{\mathsf{sim}} \wedge K \neq \perp \wedge [u] = [\mathbf{y}^\top \mathbf{k}] + K$, 0 otherwise.	
		$\mathcal{T}_{\mathsf{ver}} := \mathcal{T}_{\mathsf{ver}} \cup \{\tau\}$; $\mathcal{Q}_{\mathrm{ver}} := \mathcal{Q}_{\mathrm{ver}} \cup \{(\tau, [\mathbf{y}], \pi)\}$	

Fig. 13. Games G_0, G_1 for the proof of Theorem 4. In each procedure, the components inside a solid frame are only present in the games marked by a solid frame.

Lemma 22. *There exists an adversary $\mathcal{B}$ such that $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A}) + (Q_{\mathsf{sim}} + Q_{\mathsf{ver}}) \cdot \mathsf{poly}(\lambda)$ and*

$$|\mathbf{Adv}_0 - \mathbf{Adv}_1| \leq 2\mathbf{Adv}^{\mathsf{ind\text{-}pca}}_{\mathsf{KEM}_{\mathsf{PCA}}}(\mathcal{B}),$$

where Q_{sim}, Q_{ver}, is the number of times $\mathcal{A}$ queries SimO, VerO, respectively, and $\mathsf{poly}(\lambda)$ is independent of $\mathbf{T}(\mathcal{A})$.

Here, we use the PCA-security of $\mathsf{KEM}_{\mathsf{PCA}}$ to change the distribution of the simulated proofs.

Proof of Lemma 22. In G_1, we switch the distribution of the value of $[u]$ computed by SimO to a uniformly random element, using the IND-PCA security of $\mathsf{KEM}_{\mathsf{PCA}}$.

We build adversary $\mathcal{B}$ as follows.

Setup: $\mathcal{B}$ calls the Setup oracle for the multi-ciphertext PCA security game (see Definition 6), and gets pk, the public key of the KEM, which contains the description of a prime-order group $\mathcal{G}$. Then, $\mathcal{B}$ samples $\mathbf{M} \leftarrow_{\mathrm{R}} \mathcal{D}'_{\mathsf{par}}$ (recall that $\mathcal{D}_{\mathsf{par}}$ is WS, thus, $[\mathbf{M}]$ follows distribution $\mathcal{D}_{\mathsf{par}}$), $\mathbf{k} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^n$, and returns $(\mathcal{G}, \mathsf{crs} := ([\mathbf{M}^\top \mathbf{k}], \mathsf{pk}), [\mathbf{M}])$.

$\mathsf{SimO}(\tau, [\mathbf{y}])$: $\mathcal{B}$ calls $\mathsf{EncO}(\tau)$ and gets (C, K_b) if $\tau \notin \mathcal{T}_{\mathsf{ver}} \cup \mathcal{T}_{\mathsf{sim}}$, $\perp$ otherwise. In the former case, $\mathcal{B}$ computes $[u] := [\mathbf{y}^\top \mathbf{k}] + K_b$ and returns $(C, [u])$ to $\mathcal{A}$.

$\mathsf{VerO}(\tau, [\mathbf{y}], (C, [u]))$: $\mathcal{B}$ computes $\widehat{K} := [u] - [\mathbf{y}^\top \mathbf{k}]$, and returns $\mathsf{DecO}(\tau, C, \widehat{K})$. Note that when $\tau \notin \mathcal{T}_{\mathsf{sim}}$, we have:

$$\mathsf{DecO}(\tau, C, \widehat{K}) = 1 \text{ iff } [u] = [\mathbf{y}^\top \mathbf{k}] + K, \text{ with } K = \mathsf{Dec}_{\mathsf{KEM}}(\tau, C) \neq \perp.$$

If $\tau \in \mathcal{T}_{\mathsf{sim}}$, $\mathsf{DecO}(\tau, C, \widehat{K}) = 0$ and $\mathcal{B}$ returns 0.

This way, when $b = 0$ in the IND-PCA-security game, $\mathcal{B}$ simulates the game G_0, and when $b = 1$, it simulates G_1. Note that $\mathcal{B}$ can efficiently compute $\mathbf{M}^\perp$ from $\mathbf{M}$, and therefore, it can efficiently check the winning condition of $\mathcal{A}$. Therefore, $|\mathbf{Adv}_0 - \mathbf{Adv}_1| \leq 2\mathbf{Adv}^{\mathsf{ind\text{-}pca}}_{\mathsf{KEM}_{\mathsf{PCA}}}(\mathcal{B})$. □

Lemma 23. $\mathbf{Adv}_1 \leq \frac{Q_{\mathsf{ver}}}{q}$, *where Q_{ver} is the number of times $\mathcal{A}$ queries VerO.*

Proof of Lemma 23. We bound $\mathbf{Adv}_1$ via an information-theoretic argument. We introduce intermediate games $G_{1.j}$, for $j = 0, \ldots, Q_{\mathsf{ver}}$, where Q_{ver} is the number of times $\mathcal{A}$ queries VerO, defined

as follows: VerO is as in G_1 except that for the first j times it is queried, it outputs 0 to any input containing $[\mathbf{y}] \notin \rho$. SimO is as in G_1.

We show that:

$$\mathrm{G}_1 \equiv \mathrm{G}_{1.0} \approx_s \mathrm{G}_{1.1} \approx_s \ldots \approx_s \mathrm{G}_{1.Q_{\mathsf{ver}}}$$

where we denote statistical closeness with $\approx_s$ and statistical equality with $\equiv$.

It suffices to show that for all $j = 0, \ldots, Q_{\mathsf{ver}} - 1$, in $\mathrm{G}_{1.j}$, VerO outputs 0 to its $j+1$-st query, with overwhelming probability $1 - 1/q$ over the random coins of Setup (this implies $\mathrm{G}_{1.j} \approx_s \mathrm{G}_{1.j+1}$, with statistical difference $1/q$).

The intuition is that the vector $\mathbf{k}$ in vk contains some entropy that is hidden to the adversary. Indeed, in G_1, each simulated proof $(C, [u])$ leaks no information about $\mathbf{k}$, since $[u]$ is uniformly random. More formally, we use the fact that $\mathbf{k} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^n$ is identically distributed to $\mathbf{k} + \mathbf{M}^{\perp}\mathbf{w}$, where $\mathbf{k} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^n$, $\mathbf{w} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{n-t}$, and $\mathbf{M}^{\perp} \leftarrow_{\mathrm{R}} \mathcal{U}_{n,n-t}$ such that $\mathbf{M}^{\top}\mathbf{M}^{\perp} = \mathbf{0}$. We show that $\mathbf{w}$ is completely hidden from $\mathcal{A}$, up to its $j+1$-st query to VerO.

- The crs contains no information about $\mathbf{w}$, since $\mathbf{M}^{\top}(\mathbf{k} + \mathbf{M}^{\perp}\mathbf{w}) = \mathbf{M}^{\top}\mathbf{k}$.
- For all simulated proofs $(C, [u])$, $u \leftarrow_{\mathrm{R}} \mathbb{Z}_q$ is independent from $\mathbf{k}$.
- The first j times it is queried, VerO outputs 0, independently of its input, by definition of $\mathrm{G}_{1.j}$.

Suppose that the $j+1$-st query of $\mathcal{A}$ to VerO is: $(\tau^\star, [\mathbf{y}^\star], \pi^\star)$ such that $\mathsf{VerO}(\tau^\star, [\mathbf{y}^\star], \pi^\star) = 1 \wedge [\mathbf{y}^\star] \notin \mathcal{L}_{\mathbf{M}}$. This implies that $\pi^\star := (C^\star, [u^\star])$ is such that $[u^\star] = [\mathbf{y}^{\star\top}(\mathbf{k} + \mathbf{M}^{\perp}\mathbf{w})] + K^\star$ where $K^\star := \mathsf{Dec}_{\mathsf{KEM}}(\mathsf{sk}, \tau, C^\star)$; and $\mathbf{y}^{\star\top}\mathbf{M}^{\perp} \neq \mathbf{0}$.

This means that $\mathcal{A}$ has to guess the uniformly random value

$$[\mathbf{y}^{\star\top}\mathbf{k} + \underbrace{\mathbf{y}^{\star\top}\mathbf{M}^{\perp}\mathbf{w}}_{\neq 0, \text{ independent from } \mathcal{A}\text{'s view}}] + K^\star$$

in order to win the game, which succeeds with probability $1/q$ over the random coins of Setup. □

This completes the proof of Theorem 4. □

6.3 Tightly secure, QANIZK argument for linear subspace, with unbounded simulation soundness

In this section, we show how to adapt the DVQANIZK argument for linear subspace presented in Section 6.2, to a (publicly verifiable) QANIZK argument. The intuition behind our QANIZK construction is as follows. We use a (non-generic) technique from [22] to upgrade a DVQANIZK with unbounded simulation soundness to a QANIZK with unbounded simulation soundness using the Kernel Diffie-Hellman Assumption over pairing groups. Applying this technique to the DVQANIZK of Section 6.2 already leads to a QANIZK but as the above transformation only requires unbounded simulation soundness (in contrast to strong unbounded simulation soundness) we can apply the transformation to a simplified DVQANIZK leading to considerable efficiency improvements.

In Section 6.3, we recall the definition of pairing groups and recall the definition of the Kernel-Diffie-Hellman Assumption [28], which is the computational analogue of the MDDH Assumption. In Section 6.3, we give a QANIZK argument for linear languages.

Pairing groups. Let GGen be a probabilistic polynomial time (PPT) algorithm that on input 1^λ returns a description $\mathcal{PG} = (\mathbb{G}_1, \mathbb{G}_2, q, P_1, P_2)$ of asymmetric pairing groups where $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ are cyclic group of order q for a λ-bit prime q, P_1 and P_2 are generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is an efficiently computable (non-degenerate) bilinear map. Define $P_T := e(P_1, P_2)$, which is a generator of $\mathbb{G}_T$. We again use implicit representation of group elements. For $s \in 1, 2, T$ and $a \in \mathbb{Z}_q$, define $[a]_s = aP_s \in \mathbb{G}_s$ as the implicit representation of a in G_s . Given $[a]_1$, $[a]_2$, one can efficiently compute $[ab]_T$ using the pairing e. For two matrices $\mathbf{A}$, $\mathbf{B}$ with matching dimensions define $e([\mathbf{A}]_1, [\mathbf{B}]_2) := [\mathbf{AB}]_T \in \mathbb{G}_T$.

The Kernel-Diffie-Hellman assumption $\mathcal{D}_k$-KerMDH [28] is a natural *computational analogue* of the $\mathcal{D}_k$-MDDH Assumption.

Definition 10 ($\mathcal{D}_k$-Kernel Diffie-Hellman Assumption $\mathcal{D}_k$-KerMDH). *Let $\mathcal{D}_k$ be a matrix distribution and $s \in \{1, 2\}$. We say that the $\mathcal{D}_k$-Kernel Diffie-Hellman ($\mathcal{D}_k$-KerMDH) Assumption holds relative to GGen in group $\mathbb{G}_s$ if for all PPT adversaries $\mathcal{A}$,*

$$\mathbf{Adv}^{\mathrm{kmdh}}_{\mathcal{D}_k,\mathsf{GGen}}(\mathcal{A}) := \Pr[\mathbf{c}^\top \mathbf{A} = \mathbf{0} \wedge \mathbf{c} \neq \mathbf{0} \mid [\mathbf{c}]_{3-s} \leftarrow_{\mathrm{R}} \mathcal{A}(\mathcal{G}, [\mathbf{A}]_s)] = \mathsf{negl}(\lambda),$$

where the probability is taken over $\mathcal{G} \leftarrow_{\mathrm{R}} \mathsf{GGen}(1^\lambda)$, $\mathbf{A} \leftarrow_{\mathrm{R}} \mathcal{D}_k$.

Note that we can use a non-zero vector in the kernel of $\mathbf{A}$ to test membership in the column space of $\mathbf{A}$. This means that the $\mathcal{D}_k$-KerMDH assumption is a relaxation of the $\mathcal{D}_k$-MDDH assumption, as captured in the following lemma from [28].

Lemma 24 ([28]). *For any matrix distribution $\mathcal{D}_k$, $\mathcal{D}_k$-$\mathsf{MDDH} \Rightarrow \mathcal{D}_k$-$\mathsf{KerMDH}$.*

Our construction. In this section we describe a Tightly-secure, Quasi-adaptive Non-Interactive Zero Knowledge Argument for linear spaces with unbounded simulation soundness (see Definition 8).

We use $\mathsf{Gen}_{\mathsf{par}} = \mathsf{GGen}$. That is, $\mathsf{Gen}_{\mathsf{par}}(1^\lambda)$ returns $\mathsf{par} = \mathcal{PG}$, where $\mathcal{PG} = (\mathbb{G}_1, \mathbb{G}_2, q, P_1, P_2)$ describes asymmetric pairing groups (see Section 6.3). The probability distribution $\mathcal{D}_{\mathsf{par}}$ returns a matrix $\rho = [\mathbf{M}]_1 \in \mathbb{G}_1^{n \times t}$, for integers $n > t$. We again consider the case of *witness sampleable* (WS) distributions, see Section 6.2. Given par and ρ, the language $\mathcal{L}_{\mathbf{M}}$ is defined as

$$\mathcal{L}_{\mathbf{M}} = \Big\{ [\mathbf{y}]_1 \in \mathbb{G}_1^n : \exists\, \mathbf{x} \in \mathbb{Z}_q^t \text{ s.t. } \mathbf{y} = \mathbf{M}\mathbf{x} \Big\}.$$

Our QANIZK construction is given in Figure 14.

Theorem 5. *The protocol Π_{uss} defined in Figure 14 has perfect completeness and perfect zero-knowledge. Suppose in addition that the distribution of the matrix $\mathbf{M}$ is witness sampleable. Then, under the $\mathcal{D}_k$-MDDH Assumption in $\mathbb{G}_1$, and the $\mathcal{D}_k$-KerMDH Assumption in $\mathbb{G}_2$, the protocol has adaptive unbounded simulation soundness (see Definition 8). Namely, for any adversary $\mathcal{A}$, there exist adversaries $\mathcal{B}$ and $\mathcal{C}$ such that $\mathbf{T}(\mathcal{C}) \approx \mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A}) + Q_{\mathsf{sim}} \cdot \mathsf{poly}(\lambda)$ such that*

$$\mathbf{Adv}^{\mathrm{uss}}_{\Pi_{\mathsf{uss}}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{kmdh}}_{\mathcal{D}_k,\mathsf{GGen}}(\mathcal{B}) + 4\lambda \mathbf{Adv}^{\mathrm{mddh}}_{\mathcal{D}_k,\mathsf{GGen}}(\mathcal{C}) + 2^{-\Omega(\lambda)},$$

where Q_{sim} is the number of times $\mathcal{A}$ queries SimO, and $\mathsf{poly}(\lambda)$ is independent of $\mathbf{T}(\mathcal{A})$.

$\underline{\mathsf{Gen}(\mathsf{par}, [\mathbf{M}]_1 \in \mathbb{G}_1^{n\times t})}$:
$\mathbf{A}, \mathbf{B} \leftarrow_{\mathrm{R}} \mathcal{D}_k$
$\mathbf{K} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{n\times(k+1)}$
$\mathbf{K}_{1,0}, \ldots, \mathbf{K}_{\lambda,1} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{k\times(k+1)}$
$\mathsf{crs} := \Big([\mathbf{A}]_2, [\mathbf{K}\mathbf{A}]_2, [\overline{\mathbf{B}}]_1, [\mathbf{M}^\top\mathbf{K}]_1,$
$\big([\mathbf{K}_{j,b}\mathbf{A}]_1, [\overline{\mathbf{B}}^\top\mathbf{K}_{j,b}]_1\big)_{1\le j\le\lambda, 0\le b\le 1}\Big)$
$\mathsf{trap} := \mathbf{K}$
Return $(\mathsf{crs}, \mathsf{trap})$
//crs defines tag-space $\mathcal{T} = \{0,1\}^\lambda$

$\underline{\mathsf{Prove}(\mathsf{crs}, \tau, [\mathbf{y}]_1, \mathbf{x})}$: // $\mathbf{y} = \mathbf{M}\mathbf{x}$
$\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k; \mathbf{t} := \overline{\mathbf{B}}\mathbf{r}$
$\mathbf{u} := \mathbf{x}^\top \cdot \mathbf{M}^\top\mathbf{K} + \mathbf{r}^\top \cdot \sum_{j=1}^{\lambda} \overline{\mathbf{B}}^\top\mathbf{K}_{j,\tau_j}$
Return $\pi := ([\mathbf{t}]_1, [\mathbf{u}]_1) \in \mathbb{G}_1^k \times \mathbb{G}_1^{1\times(k+1)}$

$\underline{\mathsf{Verify}(\mathsf{crs}, \tau, [\mathbf{y}], \pi)}$:
Parse $\pi = ([\mathbf{t}]_1, [\mathbf{u}]_1)$
Compute $\mathbf{K}_\tau := \sum_{j=1}^{\lambda} \mathbf{K}_{j,\tau_j}$
Check: $e([\mathbf{u}]_1, [\mathbf{A}]_2) = e([\mathbf{y}^\top]_1, [\mathbf{K}\mathbf{A}]_2) + e([\mathbf{t}^\top]_1, [\mathbf{K}_\tau\mathbf{A}]_2)$

$\underline{\mathsf{Sim}(\mathsf{crs}, \mathsf{trap} = \mathbf{K}, \tau, [\mathbf{y}]_1)}$:
$\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k; \mathbf{t} := \overline{\mathbf{B}}\mathbf{r}$
$\mathbf{u} := \mathbf{y}^\top \cdot \mathbf{K} + \mathbf{r}^\top \cdot \sum_{j=1}^{\lambda} \overline{\mathbf{B}}^\top\mathbf{K}_{j,\tau_j}$
Return $\pi := ([\mathbf{t}]_1, [\mathbf{u}]_1) \in \mathbb{G}_1^k \times \mathbb{G}_1^{1\times(k+1)}$

Fig. 14. QANIZK argument Π_{uss} with (adaptive) unbounded simulation-soundness for WS distributions under the $\mathcal{D}_k$-MDDH Assumption and tag-space $\mathcal{T} = \{0,1\}^\lambda$.

Proof of Theorem 5. Perfect completeness and perfect zero-knowledge follow readily from the fact that for all $\mathbf{x} \in \mathbb{Z}_q^t$ and $\mathbf{y} = \mathbf{M}\mathbf{x}$, for all $\mathbf{K} \in \mathbb{Z}_q^{n\times(k+1)}$:

$$\mathbf{x}^\top(\mathbf{M}^\top\mathbf{K}) = \mathbf{y}^\top\mathbf{K}.$$

We proceed to establish adaptive unbounded simulation soundness. We show that for any adversary $\mathcal{A}$ against the simulation soundness, there exist adversaries $\mathcal{B}$ and $\mathcal{C}$ such that $\mathbf{T}(\mathcal{C}) \approx \mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A}) + Q_{\mathsf{sim}} \cdot \mathsf{poly}(\lambda)$, and

$$\mathbf{Adv}^{\mathrm{uss}}_{\Pi_{\mathsf{uss}}}(\mathcal{A}) \le \mathbf{Adv}^{\mathrm{kmdh}}_{\mathcal{D}_k,\mathsf{GGen}}(\mathcal{B}) + 4\lambda \cdot \mathbf{Adv}^{\mathrm{mddh}}_{\mathcal{D}_k,\mathsf{GGen}}(\mathcal{C}) + 2^{-\Omega(\lambda)},$$

where Q_{sim} is the number of times $\mathcal{A}$ queries SimO and $\mathsf{poly}(\lambda)$ is independent of $\mathbf{T}(\mathcal{A})$.

We proceed via a series of games and we use $\mathbf{Adv}_i$ to denote the advantage of $\mathcal{A}$ in Game G_i.

Lemma 25 (G_0 to G_1). *There exists an adversary $\mathcal{B}$ such that $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A}) + Q_{\mathsf{sim}} \cdot \mathsf{poly}(\lambda)$, and*

$$|\mathbf{Adv}_0 - \mathbf{Adv}_1| \le \mathbf{Adv}^{\mathrm{kmdh}}_{\mathcal{D}_k,\mathsf{GGen}}(\mathcal{B}),$$

where Q_{sim} is the number of times $\mathcal{A}$ queries SimO and $\mathsf{poly}(\lambda)$ is independent of $\mathbf{T}(\mathcal{A})$.

Here, we use the Kernel Diffie-Hellman Assumption to change the oracle VerO.

Proof of Lemma 25. To bound $|\mathbf{Adv}_0 - \mathbf{Adv}_1|$, it suffices to bound the probability that $\mathcal{A}$ produces $(\tau, [\mathbf{y}]_1, ([\mathbf{t}]_1, [\mathbf{u}]_1))$ that passes VerO in G_0 but not in G_1. We may rewrite the verification equation in G_0 as

$$e([\mathbf{u}]_1, [\mathbf{A}]_2) = e([\mathbf{y}^\top]_1, [\mathbf{K}'_\tau\mathbf{A}]_2) + e([\mathbf{t}^\top]_1, [\mathbf{K}_\tau\mathbf{A}]_2) \Leftrightarrow e([\mathbf{u}]_1 - [\mathbf{y}^\top\mathbf{K}'_\tau]_1 - [\mathbf{t}^\top\mathbf{K}_\tau]_1, [\mathbf{A}]_2) = \mathbf{0}$$

Observe that for any $(\tau, [\mathbf{y}]_1, ([\mathbf{t}]_1, [\mathbf{u}]_1))$ that passes verification equation in G_0 but not in G_1 the value

$$[\mathbf{u}]_1 - [\mathbf{y}^\top\mathbf{K}'_\tau]_1 - [\mathbf{t}^\top\mathbf{K}_\tau]_1$$

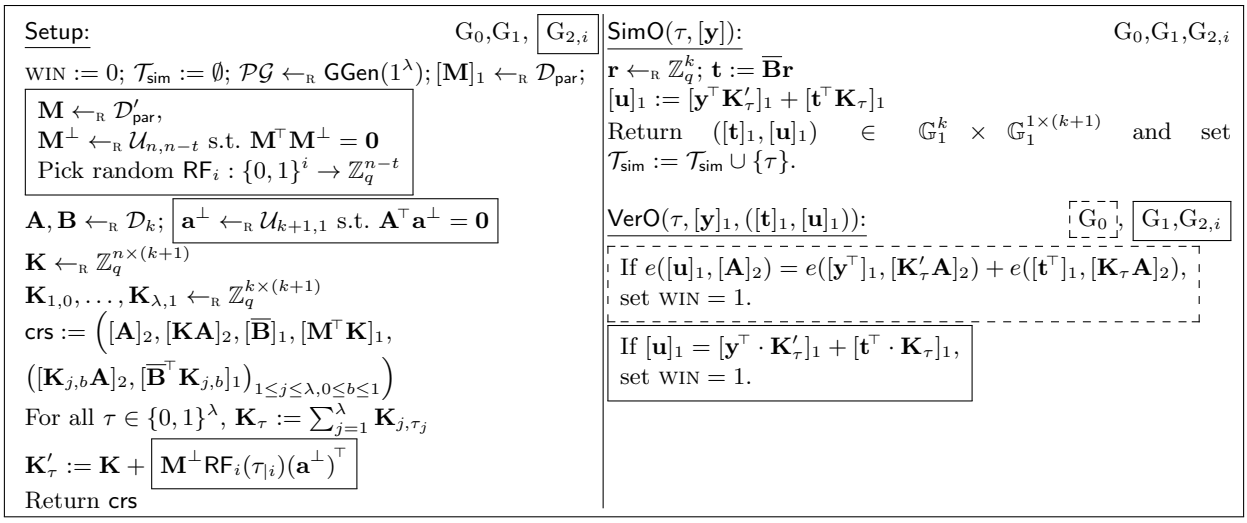

Fig. 15. Games $\mathrm{G}_0, \mathrm{G}_1, \mathrm{G}_{2,i} (0 \le i \le \lambda)$ for the proof of Theorem 5. Here, $\tau_{|i}$ denotes the i-bit prefix of τ. In each procedure, a solid (dotted) frame indicates that the command is only executed in the game marked by a solid (dotted) frame.

is a non-zero vector in the kernel of $\mathbf{A}$, which is hard to sample under the $\mathcal{D}_k$-KerMDH assumption. This means that

$$|\mathbf{Adv}_0 - \mathbf{Adv}_1| \le \mathbf{Adv}^{\mathrm{kmdh}}_{\mathcal{D}_k,\mathsf{GGen}}(\mathcal{B}).$$

□

Lemma 26 (G_1 **to** $\mathrm{G}_{2.0}$)**.**

$$\mathbf{Adv}_1 = \mathbf{Adv}_{2.0}.$$

Proof of Lemma 26. We show that the two games are statistically equivalent. To go from G_1 to $\mathrm{G}_{2.0}$, we change the distribution of $\mathbf{K} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{nk\times(k+1)}$ to $\mathbf{K} + \mathbf{M}^\perp \mathsf{RF}_0(\varepsilon)(\mathbf{a}^\perp)^\top$, where $\mathbf{K} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{n\times(k+1)}$, $\mathsf{RF}_0(\varepsilon) \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{n-t}$, $\mathbf{M}^\perp \leftarrow_{\mathrm{R}} \mathcal{U}_{n,n-t}$ such that $\mathbf{M}^\top\mathbf{M}^\perp = \mathbf{0}$, and $\mathbf{a}^\perp \leftarrow_{\mathrm{R}} \mathcal{U}_{k+1,k}$ such that $\mathbf{A}^\top\mathbf{a}^\perp$. Note that the extra term $\mathbf{M}^\perp \mathsf{RF}_0(\varepsilon)(\mathbf{a}^\perp)^\top$ does not appear in crs, since $\mathbf{M}^\top(\mathbf{K} + \boxed{\mathbf{M}^\perp \mathsf{RF}_0(\varepsilon)(\mathbf{a}^\perp)^\top}) = \mathbf{M}^\top\mathbf{K}$, and $(\mathbf{K} + \boxed{\mathbf{M}^\perp \mathsf{RF}_0(\varepsilon)(\mathbf{a}^\perp)^\top})\mathbf{A} = \mathbf{KA}$. □

Lemma 27 ($\mathrm{G}_{2.i}$ **to** $\mathrm{G}_{2.i+1}$)**.** *For all $0 \le i \le \lambda - 1$, there exists an adversary $\mathcal{B}_{2.i}$ such that $\mathbf{T}(\mathcal{B}_{2.i}) \approx \mathbf{T}(\mathcal{A}) + Q_{\mathsf{sim}} \cdot \mathsf{poly}(\lambda)$ and*

$$|\mathbf{Adv}_{2.i} - \mathbf{Adv}_{2.i+1}| \le 4 \cdot \mathbf{Adv}^{\mathrm{mddh}}_{\mathcal{D}_k,\mathsf{GGen}}(\mathcal{B}_{2.i}) + \frac{4}{q-1},$$

where Q_{sim} is the number of times $\mathcal{A}$ queries SimO*, and* poly(λ) *is independent of* $\mathbf{T}(\mathcal{A})$*.*

Overview of the proof of Lemma 27: Here, we use the $\mathcal{D}_k$-MDDH Assumption to increase the entropy in the simulated proofs, to move from RF_i to RF_{i+1}. We argue that these two games are computationally indistinguishable similarly than in Lemma 3 in [10], or Lemma 3.5 in [7]. Roughly,

the idea is to build an adversary $\mathcal{B}_{2.i}$ against the $\mathcal{D}_k$-MDDH Assumption, that guesses the value β of the $i+1$-st bit of the tag contained in $\mathcal{A}$'s query to VerO, and program the matrix $\mathbf{K}_{i+1,1-\beta}$ to embed an MDDH challenge in the simulated proofs. This way, the entropy of all simulated proof for a tag τ such that $\tau_{i+1} = 1-\beta$ increases. Formally, we use $\mathsf{RF}_{i+1} : \{0,1\}^{i+1} \to \mathbb{Z}_q^{n-t}$, defined by

$$\mathsf{RF}_{i+1}(\tau_{|i+1}) := \begin{cases} \mathsf{RF}_i(\tau_{|i}) & \text{if } \tau_{i+1} = \beta \\ \mathsf{RF}_i(\tau_{|i}) + \mathsf{RF}'_i(\tau_{|i}) & \text{if } \tau_{i+1} = 1-\beta, \end{cases}$$

where $\mathsf{RF}'_i : \{0,1\}^i \to \mathbb{Z}_q^{n-t}$ is a random function independent from RF_i.

The $\mathcal{D}_k$-MDDH Assumption tells us that we can switch a vector $[\mathbf{Br}]_1$ in the span of some rank k matrix $[\mathbf{B}]_1$ to a uniformly random vector $[\mathbf{w}]_1$ in $\mathbb{G}_1^{k+1}$. However, to go from RF_i to RF_{i+1}, we need to switch vectors $[\mathbf{Br}]_1$ to vectors with higher entropy, that are neither uniform nor in the span of $[\mathbf{B}]_1$, but of the form $[\mathbf{Br}+\mathbf{d}]_1$, where $[\mathbf{d}]_1$ is an arbitrary vector (whose distribution is neither uniform over $\mathbb{G}_1^{k+1}$, nor uniform over $\mathsf{span}([\mathbf{B}]_1)$. Therefore, we apply the $\mathcal{D}_k$-MDDH Assumption twice: once to change these vectors to uniformly random, and once again to change them to vectors of the form $[\mathbf{Br}+\mathbf{d}]_1$, where $[\mathbf{d}]_1$ is arbitrarily chosen.

Proof of Lemma 27. We build an adversary $\mathcal{B}'_{2.i}$ against the following assumption: $\mathcal{B}'_{2.i}$ receives the description of a pairing group $\mathcal{PG}$ together with a matrix $[\mathbf{B}]_1$, where $\mathbf{B} \leftarrow_{\mathrm{R}} \mathcal{D}_k$. Then $\mathcal{B}'_{2.i}$ has access to an oracle $\mathcal{O}_{\mathsf{2MDDH}}$, that takes as input a vector $[\mathbf{d}]_1 \in \mathbb{G}_1^{k+1}$ and sends back either

$$\text{Case 1: } [\mathbf{h}]_1 = [\mathbf{Bu}]_1 \text{ or Case 2: } [\mathbf{h}]_1 = [\mathbf{Bu}+\mathbf{d}]_1,$$

where $\mathbf{u} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$.

As explained in the overview, if $\mathcal{B}'_{2.i}$ calls $\mathcal{O}_{\mathsf{2MDDH}}$ at most Q_{sim} times, this assumption reduces to the Q_{sim}-fold $\mathcal{D}_k$-MDDH with a security loss of 2. Roughly, to prove this reduction, we apply the Q_{sim}-fold $\mathcal{D}_k$-MDDH Assumption twice as follows:

$$[\mathbf{Bu}]_1 \approx_{\mathsf{MDDH}} [\mathbf{v}]_1 \equiv [\mathbf{v}+\mathbf{d}]_1 \approx_{\mathsf{MDDH}} [\mathbf{Bu}+\mathbf{d}]_1,$$

where $\mathbf{u} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$, $\mathbf{v} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{k+1}$, $[\mathbf{d}]_1$ is an efficiently computable vector, $\approx_{\mathsf{MDDH}}$ denotes computational indistinguishability under the Q_{sim}-fold $\mathcal{D}_k$-MDDH Assumption, and $\equiv$ denotes statistical equivalence.

Upon receiving $\mathcal{PG}$, and $[\mathbf{B}]_1 \in \mathbb{G}_1^{k+1\times k}$, $\mathcal{B}'_{2.i}$, does the following:

- Setup :
  $\mathcal{B}'_{2.i}$ sets $\textsc{win} := 0$, $\mathcal{T}_{\mathsf{sim}} := \emptyset$, $\mathbf{M} \leftarrow_{\mathrm{R}} \mathcal{D}'_\rho$, $\mathbf{M}^\perp \leftarrow_{\mathrm{R}} \mathcal{U}_{n,n-t}$ such that $\mathbf{M}^\top\mathbf{M}^\perp = \mathbf{0}$, $\mathbf{A} \leftarrow_{\mathrm{R}} \mathcal{D}_k$, $\mathbf{a}^\perp \leftarrow_{\mathrm{R}} \mathcal{U}_{k+1,1}$ such that $\mathbf{A}^\top\mathbf{a}^\perp = \mathbf{0}$, $\mathbf{K} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{n\times(k+1)}$. Then, it picks $\beta \leftarrow_{\mathrm{R}} \{0,1\}$, which is a guess for $\tau^\star_{i+1}$, the $i+1$-st bit of the tag $\tau^\star$ of $\mathcal{A}$'s query to VerO. For all $(j,b) \neq (i+1, 1-\beta)$, it picks $\mathbf{K}_{j,b} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{k\times(k+1)}$. It also picks $\widehat{\mathbf{K}} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{k\times(k+1)}$, and implicitly defines

  $$\mathbf{K}_{i+1,1-\beta} := \widehat{\mathbf{K}} + \overline{\mathbf{B}}^{\top -1}\mathbf{B}^\top\mathbf{e}_{k+1}(\mathbf{a}^\perp)^\top,$$

  where $\mathbf{e}_{k+1}$ is the $k+1$-st vector of the canonical basis of $\mathbb{Z}_q^{k+1}$. Finally, it returns

  $$\mathsf{crs} := \Big([\mathbf{A}]_2, [\mathbf{KA}]_2, [\overline{\mathbf{B}}]_1, [\mathbf{M}^\top\mathbf{K}]_1, \big([\mathbf{K}_{j,b}\mathbf{A}]_2, [\overline{\mathbf{B}}^\top\mathbf{K}_{j,b}]_1\big)_{(j,b)\neq(i,1-\beta)},$$
  $$[\mathbf{K}_{i+1,1-\beta}\mathbf{A}]_2 = [\widehat{\mathbf{K}}\mathbf{A}]_2, [\overline{\mathbf{B}}^\top\mathbf{K}_{i,1-b}]_1 = [\overline{\mathbf{B}}^\top\widehat{\mathbf{K}} + \mathbf{B}^\top\mathbf{e}_{k+1}(\mathbf{a}^\perp)^\top]_1\Big).$$

– $\mathsf{SimO}(\tau, [\mathbf{y}]_1)$: to simulate the ρ'th query, for $\rho = 1, \ldots, Q_{\mathsf{sim}}$, $\mathcal{B}'_{2.i}$ does as follows:

If $\tau_{i+1} = \beta$ **:** $\mathcal{B}'_{2.i}$ defines on the fly $\mathsf{RF}_i(\tau_{|i})$ where $\mathsf{RF}_i : \{0,1\}^i \to \mathbb{Z}_q^{n-t}$ is a random function, and $\tau_{|i}$ denotes the i-bit prefix of τ (see Figure 15). Then it computes

$$\boxed{\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k;\ [\mathbf{t}]_1 := [\overline{\mathbf{B}} \cdot \mathbf{r}]_1}$$

$\mathbf{K}'_\tau := \mathbf{K} + \mathbf{M}^\perp \mathsf{RF}_i(\tau_{|i})(\mathbf{a}^\perp)^\top$; $\mathbf{K}_\tau := \sum_{j=1}^{\lambda} \mathbf{K}_{j,\tau_j}$ (note that $\mathcal{B}'_{2.i}$ knows the $\mathbf{K}_{j,\tau_j}$ explicitly, since $\tau_{i+1} \neq 1-\beta$); $[\mathbf{u}]_1 := [\mathbf{y}^\top \cdot \mathbf{K}'_\tau]_1 + [\mathbf{t}^\top \mathbf{K}_\tau]_1$. It returns $([\mathbf{t}]_1, [\mathbf{u}]_1)$ to $\mathcal{A}$.

If $\tau_{i+1} = 1-\beta$ **:** $\mathcal{B}'_{2.i}$ defines on the fly $\mathsf{RF}_i(\tau_{|i})$ and $\mathsf{RF}'_i(\tau_{|i})$, where $\mathsf{RF}'_i : \{0,1\}^i \to \mathbb{Z}_q^{n-t}$ is a random function independent of RF_i.
Then it sends $[\mathbf{d}_\rho]_1 := [\mathbf{y}^\top \mathbf{M}^\perp \mathsf{RF}'_i(\tau_{|i}) \cdot \mathbf{e}_{k+1}]_1$ to its oracle $\mathcal{O}_{\mathsf{2MDDH}}$ to get back

$$\text{Case 1: } [\mathbf{h}_\rho]_1 := [\mathbf{B}\mathbf{w}_\rho]_1 \text{ or Case 2: } [\mathbf{h}_\rho]_1 := [\mathbf{B}\mathbf{w}_\rho + \mathbf{y}^\top \mathbf{M}^\perp \mathsf{RF}'_i(\tau_{|i}) \cdot \mathbf{e}_{k+1}]_1,$$

Then it sets

$$\boxed{\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k;\ [\mathbf{t}]_1 := [\overline{\mathbf{B}}\mathbf{r}]_1 + [\overline{\mathbf{h}}_\rho]_1}$$

$\mathbf{K}'_\tau := \mathbf{K} + \mathbf{M}^\perp \mathsf{RF}_i(\tau_{|i})(\mathbf{a}^\perp)^\top$;

$$[\mathbf{u}]_1 := [\mathbf{y}^\top \cdot \mathbf{K}'_\tau]_1 + [\mathbf{t}^\top \sum_{j \neq i+1} \mathbf{K}_{j,\tau_j} + \mathbf{t}^\top \widehat{\mathbf{K}} + (\mathbf{B}\mathbf{r} + \mathbf{h}_\rho)^\top \mathbf{e}_{k+1}(\mathbf{a}^\perp)^\top]_1$$

It returns $([\mathbf{t}]_1, [\mathbf{u}]_1)$ to $\mathcal{A}$.

– $\mathsf{VerO}(\tau^\star, [\mathbf{y}^\star]_1, ([\mathbf{t}^\star]_1, [\mathbf{u}^\star]_1))$:
If $\tau^\star_{i+1} \neq \beta$, abort. Otherwise, defines on the fly $\mathsf{RF}_i(\tau^\star_{|i})$; computes $\mathbf{K}'_{\tau^\star} := \mathbf{K} + \mathbf{M}^\perp \mathsf{RF}_i(\tau^\star_{|i})(\mathbf{a}^\perp)^\top$; $\mathbf{K}_{\tau^\star} := \sum_{j=1}^{\lambda} \mathbf{K}_{j,\tau^\star_j}$ (note that $\mathcal{B}'_{2.i}$ knows the $\mathbf{K}_{j,\tau^\star_j}$ explicitly since $\tau^\star_{i+1} = \beta \neq 1-\beta$) and sets $\text{WIN} = 1$ if the following is true:

$$\mathbf{y}^\top \cdot \mathbf{M}^\perp \neq \mathbf{0} \wedge \tau \notin \mathcal{T}_{\mathsf{sim}} \wedge [\mathbf{u}]_1 = [\mathbf{y}^\top \mathbf{K}'_{\tau^\star}]_1 + [\mathbf{t}^\top \mathbf{K}_{\tau^\star}]_1$$

Let us analyze the simulation of Setup, SimO, and VerO by $\mathcal{B}'_{2.i}$. We show that if $\mathcal{B}'_{2.i}$ guesses β successfully, in Case 1, it simulates $\mathrm{G}_{2.i}$, and in Case 2, it simulates $\mathrm{G}_{2.i+1}$ (recall that the cases refer to the possible output distributions of the oracle $\mathcal{O}_{\mathsf{2MDDH}}$).

First, the crs generated by $\mathcal{B}'_{2.i}$ in both Case 1 and Case 2 is distributed as in $\mathrm{G}_{2.i}$ or $\mathrm{G}_{2.i+1}$ (the crs is identically distributed in these two games), since the two following distributions are identical:

$$\mathbf{K}_{j+1,\beta} \text{ and } \widehat{\mathbf{K}} + \overline{\mathbf{B}}^{\top^{-1}} \mathbf{B}^\top \mathbf{e}_{k+1}(\mathbf{a}^\perp)^\top,$$

where $\mathbf{K}_{j+1,\beta} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{k \times (k+1)}$, $\widehat{\mathbf{K}} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{k \times (k+1)}$.

Now, let us analyze the simulation of SimO. In both Case 1 and Case 2, the vectors $[\mathbf{t}]_1$ are distributed as in $\mathrm{G}_{2.i}$ and $\mathrm{G}_{2.i+1}$ (vectors $[\mathbf{t}]_1$ are identically distributed in these two games) since the two following distribution are equivalent:

$$[\overline{\mathbf{B}} \cdot \mathbf{r}]_1,\ \mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k \text{ and } [\overline{\mathbf{B}} \cdot \mathbf{r} + \overline{\mathbf{h}}_\rho]_1,\ \mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$$

since in Case 1 and 2, we have $\overline{\mathbf{h}}_\rho := \overline{\mathbf{B}}\mathbf{w}_\rho$ with $\mathbf{w}_\rho \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$. Now we prove that in Case 1, the vectors $[\mathbf{u}]_1$ are distributed as in $\mathrm{G}_{2.i}$, and that in Case 2, they are distributed as in $\mathrm{G}_{2.i+1}$.

For queries with $\tau_{i+1} = \beta$: this is straightforward since $\mathsf{RF}_{i+1}(\tau_{|i+1}) = \mathsf{RF}_i(\tau_{|i})$, i.e, $\mathrm{G}_{2.i}$ and $\mathrm{G}_{2.i+1}$ are identically distributed for these queries.

For queries with $\tau_{i+1} = 1 - \beta$: We use the following notation:

$$[\mathbf{u}_{\setminus i+1}]_1 := [\mathbf{y}^\top(\mathbf{K} + \mathbf{M}^\perp \mathsf{RF}_i(\tau_{|i})(\mathbf{a}^\perp)^\top)]_1 + [\mathbf{t}^\top \sum_{j \neq i+1}^{\lambda} \mathbf{K}_{j,\tau_j}]_1.$$

In Case 1, $\mathcal{B}'_{2.i}$ computes:

$$\begin{aligned}
[\mathbf{u}]_1 &:= [\mathbf{u}_{\setminus i+1}]_1 + [\mathbf{t}^\top \widehat{\mathbf{K}} + \big(\mathbf{B}\mathbf{r} + \mathbf{B}\mathbf{w}_\rho\big)^\top \mathbf{e}_{k+1}(\mathbf{a}^\perp)^\top]_1 \\
&= [\mathbf{u}_{\setminus i+1}]_1 + [\mathbf{t}^\top \widehat{\mathbf{K}} + \underbrace{(\overline{\mathbf{B}}\mathbf{r} + \overline{\mathbf{B}}\mathbf{w}_\rho)^\top}_{=[\mathbf{t}^\top]_1} \overline{\mathbf{B}}^{\top -1} \mathbf{B}^\top \mathbf{e}_{k+1}(\mathbf{a}^\perp)^\top]_1 \\
&= [\mathbf{u}_{\setminus i+1}]_1 + [\mathbf{t}^\top \underbrace{\Big(\widehat{\mathbf{K}} + \overline{\mathbf{B}}^{\top -1} \mathbf{B}^\top \mathbf{e}_{k+1}(\mathbf{a}^\perp)^\top\Big)}_{=\mathbf{K}_{i+1,\tau_{i+1}}}]_1 \\
&= \underbrace{[\mathbf{u}_{\setminus i+1}]_1 + [\mathbf{t}^\top \mathbf{K}_{i+1,\tau_{i+1}}]_1}_{\text{in } \mathrm{G}_{2.i}}
\end{aligned}$$

In Case 2, $\mathcal{B}'_{2.i}$ computes:

$$\begin{aligned}
[\mathbf{u}]_1 &:= [\mathbf{u}_{\setminus i+1}]_1 + [\mathbf{t}^\top \widehat{\mathbf{K}} + \big(\mathbf{B}\mathbf{r} + \mathbf{B}\mathbf{w}_\rho + \boxed{\mathbf{y}^\top \mathbf{M}^\perp \mathsf{RF}'_i(\tau_{|i})\mathbf{e}_{k+1}}\big)^\top \mathbf{e}_{k+1}(\mathbf{a}^\perp)^\top)]_1 \\
&= [\mathbf{u}_{\setminus i+1}]_1 + [\mathbf{t}^\top \underbrace{\Big(\widehat{\mathbf{K}} + \overline{\mathbf{B}}^{\top -1} \mathbf{B}^\top \mathbf{e}_{k+1}(\mathbf{a}^\perp)^\top\Big)}_{=\mathbf{K}_{i+1,\tau_{i+1}}} + \boxed{\mathbf{y}^\top \mathbf{M}^\perp \mathsf{RF}'_i(\tau_{|i})(\mathbf{a}^\perp)^\top}]_1 \\
&= \underbrace{[\mathbf{u}_{\setminus i+1}]_1 + [\mathbf{t}^\top \mathbf{K}_{i+1,\tau_{i+1}}]_1 + \boxed{[\mathbf{y}^\top \mathbf{M}^\perp \mathsf{RF}'_i(\tau_{|i})(\mathbf{a}^\perp)^\top]_1}}_{\text{in } \mathrm{G}_{2.i+1}}
\end{aligned}$$

Finally, if $\mathcal{B}'_{2.i}$ guesses $\beta = \tau^\star_{i+1}$ correctly (this happens with probability 1/2), it simulates VerO as in $\mathrm{G}_{2.i}$ and $\mathrm{G}_{2.i+1}$ (VerO's outputs are identically distributed in these two games), in both Case 1 and Case 2.

Therefore, $\mathcal{B}'_{2.i}$, which runs in $\mathbf{T}(\mathcal{B}'_{2.i}) \approx \mathbf{T}(\mathcal{A}) + Q_{\mathsf{sim}} \cdot \mathsf{poly}(\lambda)$, with $\mathsf{poly}(\lambda)$ independent of $\mathbf{T}(\mathcal{A})$, can use adversary $\mathcal{A}$ to differentiate Case 1 from Case 2 with probability at least $\frac{|\mathbf{Adv}_{2.i} - \mathbf{Adv}_{2.i+1}|}{2}$. As we explained at the beginning of the proof, this implies the existence of an adversary $\mathcal{B}''_{2.i}$ such that $\mathbf{T}(\mathcal{B}''_{2.i}) \approx \mathbf{T}(\mathcal{A}) + Q_{\mathsf{sim}} \cdot \mathsf{poly}(\lambda)$ with $\mathsf{poly}(\lambda)$ independent of $\mathbf{T}(\mathcal{A})$, and such that

$$|\mathbf{Adv}_{2.i} - \mathbf{Adv}_{2.i+1}| \leq 4\mathbf{Adv}^{Q_{\mathsf{sim}}\text{-mddh}}_{\mathcal{D}_k,\mathsf{GGen}}(\mathcal{B}_{2.i}).$$

The factor 4 comes from security loss 2 for guessing $\beta = \tau^\star_{i+1}$, and 2 to reduce to the Q_{sim}-fold $\mathcal{D}_k$-MDDH Assumption. This implies the Lemma by random self-reducibility of $\mathcal{D}_k$-MDDH, [14]. □

Lemma 28 ($\mathrm{G}_{2.\lambda}$). $\mathbf{Adv}_{2.\lambda} \leq 1/q$.

Proof of Lemma 28. We bound $\mathbf{Adv}_{2.\lambda}$ via an information-theoretic argument. Recall that $\mathsf{VerO}(\tau^\star, [\mathbf{y}^*], \pi^* = ([\mathbf{t}^\star]_1, [\mathbf{u}^\star]_1))$ set $\textsc{win} = 1$ if the following properties are satisfied:

Property 1 : $\tau^\star \notin \mathcal{T}_{\mathsf{sim}}$
Property 2 : $\mathbf{y}^\star \notin \mathsf{span}(\mathbf{M})$
Property 3 : $[\mathbf{u}]_1 := \left[\mathbf{y}^{\star\top}\left(\mathbf{K} + \mathbf{M}^\perp \mathsf{RF}_\lambda(\tau^\star)(\mathbf{a}^\perp)^\top\right)\right]_1 + [\mathbf{t}^{\star\top}\mathbf{K}_{\tau^\star}]_1$, where $\mathsf{RF}_\lambda : \{0,1\}^\lambda \to \mathbb{Z}_q^{n-t}$ is a random function.

We show that the value $\mathbf{y}^{\star\top}\mathbf{M}^\perp \mathsf{RF}_\lambda(\tau^\star) \in \mathbb{G}_1$ is completely hidden from $\mathcal{A}$, up to its query to VerO.

We first look at what the adversary's view leaks about the value $\mathsf{RF}_\lambda(\tau^\star)$.

- The crs contains no information about $\mathsf{RF}_\lambda(\tau^\star)$.
- If $\tau^\star \notin \mathcal{T}_{\mathsf{sim}}$, then $\mathsf{RF}_\lambda(\tau^\star)$ is independent of $\{\mathsf{RF}_\lambda(\tau), \tau \in \mathcal{T}_{\mathsf{sim}}\}$, because RF_λ is a random function, and therefore, $\mathsf{RF}_\lambda(\tau^\star)$ is independent of the outputs of SimO.

Thus, if Property 1 and 2 are satisfied, the value

$$\underbrace{\mathbf{y}^{*\top}\mathbf{M}^\perp}_{\neq \mathbf{0} \text{ by Property 1}} \cdot \underbrace{\mathsf{RF}_\lambda(\tau^\star)}_{\text{uniformly random, by Property 2}}$$

is a uniformly random over $\mathbb{G}_1$ from $\mathcal{A}$'s viewpoint. Therefore, Property 3 holds with probability at most $1/q$ over the random choice of $\mathsf{RF}_\lambda(\tau^\star)$. This proves $\mathbf{Adv}_{1,\lambda} \leq 1/q$. □

Finally, Theorem 5 follows readily from Lemmas 25-28. □

Acknowledgments. We would like to thank Jie Chen for insightful and inspiring discussions. This work was done in part while the first and last authors were visiting the Simons Institute for the Theory of Computing, supported by the Simons Foundation and NSF grant CNS-1523467.

References

1. M. Abdalla, F. Benhamouda, and D. Pointcheval. Public-key encryption indistinguishable under plaintext-checkable attacks. In J. Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 332–352. Springer, Mar. / Apr. 2015.
2. M. Abe, M. Chase, B. David, M. Kohlweiss, R. Nishimaki, and M. Ohkubo. Constant-size structure-preserving signatures: Generic constructions and simple assumptions. In X. Wang and K. Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 4–24. Springer, Dec. 2012.
3. M. Abe, B. David, M. Kohlweiss, R. Nishimaki, and M. Ohkubo. Tagged one-time signatures: Tight security and optimal tag size. In K. Kurosawa and G. Hanaoka, editors, *PKC 2013*, volume 7778 of *LNCS*, pages 312–331. Springer, Feb. / Mar. 2013.
4. M. Abe, R. Gennaro, and K. Kurosawa. Tag-KEM/DEM: A new framework for hybrid encryption. *Journal of Cryptology*, 21(1):97–130, Jan. 2008.
5. N. Attrapadung, G. Hanaoka, and S. Yamada. A framework for identity-based encryption with almost tight security. Cryptology ePrint Archive, Report 2015/566, 2015. `http://eprint.iacr.org/2015/566`.
6. M. Bellare, A. Boldyreva, and S. Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In B. Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 259–274. Springer, May 2000.
7. O. Blazy, E. Kiltz, and J. Pan. (hierarchical) identity-based encryption from affine message authentication. In J. A. Garay and R. Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 408–425. Springer, Aug. 2014.
8. D. Boneh, R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. *SIAM Journal on Computing*, 36(5):1301–1328, 2007.

9. J. Camenisch, N. Chandran, and V. Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In A. Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 351–368. Springer, Apr. 2009.
10. J. Chen and H. Wee. Fully, (almost) tightly secure IBE and dual system groups. In R. Canetti and J. A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 435–460. Springer, Aug. 2013.
11. R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003.
12. A. De Santis, G. Di Crescenzo, R. Ostrovsky, G. Persiano, and A. Sahai. Robust non-interactive zero knowledge. In J. Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 566–598. Springer, Aug. 2001.
13. D. Dolev, C. Dwork, and M. Naor. Nonmalleable cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000.
14. A. Escala, G. Herold, E. Kiltz, C. Ràfols, and J. Villar. An algebraic framework for Diffie-Hellman assumptions. In R. Canetti and J. A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 129–147. Springer, Aug. 2013.
15. S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.
16. J. Gong, J. Chen, X. Dong, Z. Cao, and S. Tang. Extended nested dual system groups, revisited. Cryptology ePrint Archive, Report 2015/820, 2015. `http://eprint.iacr.org/`.
17. D. Hofheinz and T. Jager. Tightly secure signatures and public-key encryption. In R. Safavi-Naini and R. Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 590–607. Springer, Aug. 2012.
18. D. Hofheinz and E. Kiltz. Secure hybrid encryption from weakened key encapsulation. In A. Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 553–571. Springer, Aug. 2007.
19. D. Hofheinz, J. Koch, and C. Striecks. Identity-based encryption with (almost) tight security in the multi-instance, multi-ciphertext setting. In J. Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 799–822. Springer, Mar. / Apr. 2015.
20. C. S. Jutla and A. Roy. Shorter quasi-adaptive NIZK proofs for linear subspaces. In K. Sako and P. Sarkar, editors, *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 1–20. Springer, Dec. 2013.
21. E. Kiltz. Chosen-ciphertext security from tag-based encryption. In S. Halevi and T. Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 581–600. Springer, Mar. 2006.
22. E. Kiltz and H. Wee. Quasi-adaptive NIZK for linear subspaces revisited. In E. Oswald and M. Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 101–128. Springer, Apr. 2015.
23. K. Kurosawa and Y. Desmedt. A new paradigm of hybrid encryption scheme. In M. Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 426–442. Springer, Aug. 2004.
24. B. Libert, M. Joye, M. Yung, and T. Peters. Concise multi-challenge CCA-secure encryption and signatures with almost tight security. In P. Sarkar and T. Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 1–21. Springer, Dec. 2014.
25. B. Libert, T. Peters, M. Joye, and M. Yung. Non-malleability from malleability: Simulation-sound quasi-adaptive NIZK proofs and CCA2-secure encryption from homomorphic signatures. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 514–532. Springer, May 2014.
26. B. Libert, T. Peters, M. Joye, and M. Yung. Compactly hiding linear spans: Tightly secure constant-size simulation-sound QA-NIZK proofs and applications. Cryptology ePrint Archive, Report 2015/242, 2015. `http://eprint.iacr.org/2015/242`.
27. B. Libert, T. Peters, M. Joye, and M. Yung. Compactly hiding linear spans: Tightly secure constant-size simulation-sound qa-nizk proofs and applications. Cryptology ePrint Archive, Report 2015/242, 2015. `http://eprint.iacr.org/`.
28. P. Morillo, C. Ràfols, and J. L. Villar. Matrix computational assumptions in multilinear groups. *IACR Cryptology ePrint Archive*, 2015:353, 2015.
29. M. Naor and O. Reingold. Number-theoretic constructions of efficient pseudo-random functions. *Journal of the ACM*, 51(2):231–262, 2004.
30. M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd ACM STOC*, pages 427–437. ACM Press, May 1990.
31. T. Okamoto and D. Pointcheval. REACT: rapid enhanced-security asymmetric cryptosystem transform. In *Topics in Cryptology - CT-RSA 2001, The Cryptographer's Track at RSA Conference 2001, San Francisco, CA, USA, April 8-12, 2001, Proceedings*, pages 159–175, 2001.
32. C. Rackoff and D. R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In J. Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 433–444. Springer, Aug. 1992.

33. A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th FOCS*, pages 543–553. IEEE Computer Society Press, Oct. 1999.