Kurosawa-Desmedt Meets Tight Security

Romain Gay[1] *, Dennis Hofheinz[2] †, and Lisa Kohl[2] ‡

[1] Département d'informatique de l'ENS, École normale supérieure, CNRS, PSL Research University, 75005 Paris, France, and INRIA
rgay@di.ens.fr

[2] Karlsruhe Institute of Technology, Karlsruhe, Germany
Dennis.Hofheinz, Lisa.Kohl@kit.edu

Abstract. At EUROCRYPT 2016, Gay et al. presented the first pairing-free public-key encryption (PKE) scheme with an almost tight security reduction to a standard assumption. Their scheme is competitive in efficiency with state-of-the art PKE schemes and has very compact ciphertexts (of three group elements), but suffers from a large public key (of about 200 group elements).
In this work, we present an improved pairing-free PKE scheme with an almost tight security reduction to the Decisional Diffie-Hellman assumption, small ciphertexts (of three group elements), *and* small public keys (of six group elements). Compared to the work of Gay et al., our scheme thus has a considerably smaller public key and comparable other characteristics, although our encryption and decryption algorithms are somewhat less efficient.
Technically, our scheme borrows ideas both from the work of Gay et al. and from a recent work of Hofheinz (EUROCRYPT, 2017). The core technical novelty of our work is an efficient and compact designated-verifier proof system for an OR-like language. We show that adding such an OR-proof to the ciphertext of the state-of-the-art PKE scheme from Kurosawa and Desmedt enables a tight security reduction.

Keywords. Public key encryption, tight security.

1 Introduction

Tight security reductions. We are usually interested in cryptographic schemes that come with a *security reduction* to a computational assumption. A security reduction shows that every attack on the scheme can be translated into an attack on a computational assumption. Thus, the only way to break the scheme is to solve an underlying mathematical problem. We are most interested in reductions to well-investigated, "standard" assumptions, and in reductions that are "tight". A tight security reduction ensures that the reduction translates attacks on the scheme into attacks on the assumption that are of similar complexity and success probability. In other words, the difficulty of breaking the scheme is quantitatively not lower than the difficulty of breaking the investigated assumption.

Tight security reductions are also beneficial from a practical point of view. Indeed, assume that we choose the keylength of a scheme so as to guarantee that the only way to break that scheme is to break a computational assumption on currently secure parameters.[3] Then, a tight reduction enables smaller keylength recommendations (than with a non-tight reduction in which, say, the attack on the assumption is much more complex than the attack on the scheme).

*Supported by ERC Project aSCEND (639554).

†Supported by DFG grants HO 4534/4-1 and HO 4534/2-2.

‡Supported by DFG grant HO 4534/2-2.

[3]This is unfortunately different from current practice, which does not take into account security reductions at all: practical keylength recommendations are such that known attacks on the *scheme itself* are infeasible [18].

Reference	$\|pk\|$	$\|c\| - \|m\|$	sec. loss	assumption	pairing
CS98 [6]	3	3	$\mathcal{O}(Q)$	1-LIN = DDH	no
KD04, HK07 [17, 14]	$k+1$	$k+1$	$\mathcal{O}(Q)$	k-LIN ($k \geq 1$)	no
HJ12 [13]	$O(1)$	$O(\lambda)$	$\mathcal{O}(1)$	2-LIN	yes
LPJY15 [19, 20]	$\mathcal{O}(\lambda)$	47	$\mathcal{O}(\lambda)$	2-LIN	yes
AHY15 [2]	$\mathcal{O}(\lambda)$	12	$\mathcal{O}(\lambda)$	2-LIN	yes
GCDCT15 [10, 15]	$\mathcal{O}(\lambda)$	$6k$	$\mathcal{O}(\lambda)$	k-LIN ($k \geq 1$)	yes
GHKW16 [9]	$2\lambda k$	$3k$	$\mathcal{O}(\lambda)$	k-LIN ($k \geq 1$)	no
H16 [11]	$2k(k+5)$	$k+4$	$\mathcal{O}(\lambda)$	k-LIN ($k \geq 2$)	yes
H16 [11]	20	28	$\mathcal{O}(\lambda)$	DCR	—
Ours	6	3	$\mathcal{O}(\lambda)$	1-LIN = DDH	no
	$2k(k+4)$	$4k$	$\mathcal{O}(\lambda)$	k-LIN ($k \geq 2$)	no

Fig. 1: Comparison amongst CCA-secure encryption schemes, where Q is the number of ciphertexts, $|pk|$ denotes the size (in groups elements) of the public key, and $|c| - |m|$ denotes the ciphertext overhead, ignoring smaller contributions from symmetric-key encryption.

Tightly secure PKE schemes. The focus of this paper are public-key encryption (PKE) schemes with a tight security reduction. The investigation of this topic was initiated already in 2000 by Bellare, Boldyreva, and Micali [3]. However, the first tightly secure encryption scheme based on a standard assumption was presented only in 2012 [13], and was far from practical. Many more efficient schemes were proposed [1, 5, 4, 19, 15, 20, 2, 10, 12, 11] subsequently, but Gay et al. [9] (henceforth GHKW) were the first to present a pairing-free tightly secure PKE scheme from a standard assumption. Their PKE scheme has short ciphertexts (of three group elements), and its efficiency compares favorably with the popular Cramer-Shoup encryption scheme. Still, the GHKW construction suffers from a large public key (of about 200 group elements). Fig. 1 summarizes relevant features of selected existing PKE schemes.

Our contribution. In this work, we construct a pairing-free PKE scheme with an almost[4] tight security reduction to a standard assumption (the Decisional Diffie-Hellman assumption), and with short ciphertexts and keys. Our scheme improves upon GHKW in that it removes its main disadvantage (of large public keys), although our encryption and decryption algorithms are somewhat less efficient than those of GHKW.

Our construction can be seen as a variant of the state-of-the-art Kurosawa-Desmedt PKE scheme [17] with an additional consistency proof. This consistency proof ensures that ciphertexts are of a special form, and is in fact very efficient (in that it only occupies one additional group element in the ciphertext). This proof is the main technical novelty of our scheme, and is the key ingredient to enable an almost tight security reduction.

[4]Like [5], we call our reduction *almost* tight, since its loss (of λ) is independent of the number of challenges and users, but not constant.

Technical overview. The starting point of our scheme is the Kurosawa-Desmedt PKE scheme from [17]. In this scheme, public parameters, public keys, and ciphertexts are of the following form:[5]

$$\begin{aligned}
\mathit{pars} &= [\,\mathbf{A}\,] \in \mathbb{G}^{2\times 1} && \text{for random } \mathbf{A} \in \mathbb{Z}_{|\mathbb{G}|}^{2\times 1} \\
\mathit{pk} &= [\,\mathbf{k}_0^\top \mathbf{A},\, \mathbf{k}_1^\top \mathbf{A}\,] \in \mathbb{G}\times\mathbb{G} && \text{for random } \mathbf{k}_0, \mathbf{k}_1 \in \mathbb{Z}_{|\mathbb{G}|}^{2} \\
C &= \big([\,\mathbf{c} = \mathbf{A}\mathbf{r}\,],\, \mathbf{E}_K(M)\big) && \text{for random } \mathbf{r} \in \mathbb{Z}_{|\mathbb{G}|}, \\
& && K = [(\mathbf{k}_0 + \tau\mathbf{k}_1)^\top \mathbf{A}\mathbf{r}], \\
& && \text{and } \tau = H([\mathbf{c}]).
\end{aligned} \tag{1}$$

Here, $\mathbf{E}$ is the encryption algorithm of a symmetric authenticated encryption scheme, and H is a collision-resistant hash function.

In their (game-based) proof of IND-CCA security (with one scheme instance and one challenge ciphertext), Kurosawa and Desmedt proceed as follows: first, they use the secret key $\mathbf{k}_0, \mathbf{k}_1$ to generate the value K in the challenge ciphertext from a given $[\mathbf{c}] = [\mathbf{A}\mathbf{r}]$ (through $K = [(\mathbf{k}_0 + \tau\mathbf{k}_1)^\top \mathbf{c}]$). This enables the reduction to forget the witness $\mathbf{r}$, and thus to modify the distribution of $\mathbf{c}$. Next, Kurosawa and Desmedt use the Decisional Diffie-Hellman (DDH) assumption to modify the setup of $\mathbf{c}$ to a random vector not in the span of $\mathbf{A}$. Finally, they argue that this change effectively randomizes the value K from the challenge ciphertext (which then enables a reduction to the security of $\mathbf{E}$).

To see that K is indeed randomized, note that once $\mathbf{c} \notin \mathrm{span}(\mathbf{A})$, the value $K = [(\mathbf{k}_0 + \tau\mathbf{k}_1)^\top \mathbf{c}]$ depends on entropy in $\mathbf{k}_0, \mathbf{k}_1$ that is not leaked through pk. Furthermore, Kurosawa and Desmedt show that even a decryption oracle leaks no information about that entropy. (Intuitively, this holds since any decryption query with $\mathbf{c} \in \mathrm{span}(\mathbf{A})$ only reveals information about $\mathbf{k}_0, \mathbf{k}_1$ that is already contained in pk. On the other hand, any decryption query with $\mathbf{c} \notin \mathrm{span}(\mathbf{A})$ results in a computed key K that is independently random, and thus will lead the symmetric authenticated encryption scheme to reject the whole ciphertext.)

An argument of Bellare, Boldyreva, and Micali [3] (which is applied in [3] to the related Cramer-Shoup encryption scheme) shows that the security proof for the Kurosawa-Desmedt scheme carries over to a setting with many users. Due to the re-randomizability properties of the DDH assumption, the quality of the corresponding security reduction does not degrade in the multi-user scenario. The security proof of Kurosawa and Desmedt does however not immediately scale to a larger number of *ciphertexts*. Indeed, observe that the final argument to randomize K relies on the entropy in $\mathbf{k}_0, \mathbf{k}_1$. Since this entropy is limited, only a limited number of ciphertexts (per user) can be randomized at a time.[6]

First trick: randomize $\mathbf{k}_0$. In our scheme, we adapt two existing techniques for achieving tight security. The first trick, which we borrow from GHKW [9] (who in turn build upon [5, 15]), consists in modifying the secret key $\mathbf{k}_0, \mathbf{k}_1$ first, before randomizing the values K from challenge ciphertexts. Like the original Kurosawa-Desmedt proof, our argument starts out by first using $\mathbf{k}_0, \mathbf{k}_1$ to generate challenge ciphertexts, and then simultaneously randomizing all values $\mathbf{c}$ from challenges (using the

[5] In this paper, we use an implicit notation for group elements. That is, we write $[\mathbf{x}] := g^{\mathbf{x}} \in \mathbb{G}^n$ for a fixed group generator $g \in \mathbb{G}$ and a vector $\mathbf{x} \in \mathbb{Z}_{|\mathbb{G}|}^n$, see [8]. We also use the shorthand notation $[\mathbf{x}, \mathbf{y}] := ([\mathbf{x}], [\mathbf{y}])$.

[6] We note that a generic hybrid argument shows the security of the Kurosawa-Desmedt scheme in a multi-ciphertext setting. However, the corresponding security reduction loses a factor of Q in success probability, where Q is the number of challenge ciphertexts.

re-randomizability of DDH). But then we use another reduction to DDH, with the DDH challenges embedded into $\mathbf{k}_0$ *and* in all challenge $\mathbf{c}$, to simultaneously randomize all challenge K at once.

During this last reduction, we will (implicitly) set up $\mathbf{k}_0 = \mathbf{k}'_0 + \alpha\mathbf{A}^\perp$ for a known $\mathbf{k}'_0$, a known $\mathbf{A}^\perp \in \mathbb{Z}_{|\mathbb{G}|}^{2\times 1}$ with $(\mathbf{A}^\perp)^\top\mathbf{A} = \mathbf{0}$, and an unknown $\alpha \in \mathbb{Z}_{|\mathbb{G}|}$ from the DDH challenge $[\alpha, \beta, \gamma]$. We can thus decrypt all ciphertexts with $\mathbf{c} \in \mathrm{span}(\mathbf{A})$ (since $\mathbf{k}_0^\top\mathbf{Ar} = \mathbf{k}_0'^\top\mathbf{Ar}$), and randomize all challenge ciphertexts (since their $\mathbf{c}$ satisfies $\mathbf{c} \notin \mathrm{span}(\mathbf{A})$ and thus allows to embed β and γ into $\mathbf{c}$ and K, respectively). However, we will not be able to answer decryption queries with $\mathbf{c} \notin \mathrm{span}(\mathbf{A})$. Hence, before applying this trick, we will need to make sure that any such decryption query will be rejected anyway.

Second trick: the consistency proof. We do not know how to argue (with a tight reduction) that such decryption queries are rejected in the original Kurosawa-Desmedt scheme from (1). Instead, we introduce an additional consistency proof in the ciphertext, so ciphertexts in our scheme now look as follows:

$$C = \big([\,\mathbf{c} = \mathbf{Ar}\,],\ \pi,\ \mathbf{E}_K(M)\big) \quad \begin{aligned} &\text{for random } \mathbf{r} \in \mathbb{Z}_{|\mathbb{G}|}, \\ &K = [(\mathbf{k}_0 + \tau\mathbf{k}_1)^\top\mathbf{Ar}], \\ &\text{and } \tau = H([\mathbf{c}]). \end{aligned} \tag{2}$$

Here, π is a proof (yet to be described) that shows the following statement:

$$\mathbf{c} \in \mathrm{span}(\mathbf{A})\ \lor\ \mathbf{c} \in \mathrm{span}(\mathbf{A}_0)\ \lor\ \mathbf{c} \in \mathrm{span}(\mathbf{A}_1), \tag{3}$$

where $\mathbf{A}_0, \mathbf{A}_1 \in \mathbb{Z}_{|\mathbb{G}|}^{2\times 1}$ are different (random but fixed) matrices. Our challenge ciphertexts will satisfy (3) at all times, even after their randomization.

We will then show that all "inconsistent" decryption queries (with $\mathbf{c} \notin \mathrm{span}(\mathbf{A})$) are rejected with a combination of arguments from GHKW [9] and Hofheinz [11]. We will proceed in a number of hybrids. In the i-th hybrid, all challenge ciphertexts are prepared with a value of $\mathbf{k}_0 + \mathbf{F}_i(\tau_{|i})$ instead of $\mathbf{k}_0$, where $\mathbf{F}_i(\tau_{|i})$ is a random function applied to the first i bits of τ. Likewise, in all decryption queries with inconsistent $\mathbf{c}$ (i.e., with $\mathbf{c} \notin \mathrm{span}(\mathbf{A})$), we use $\mathbf{k}_0 + \mathbf{F}_i(\tau_{|i})$. Going from the i-th to the $(i+1)$-th hybrid proceeds in a way that is very similar to the one from GHKW: First, we set up the $\mathbf{c}$ value in each challenge ciphertext to be in $\mathrm{span}(\mathbf{A}_{\tau_{i+1}})$, where τ_{i+1} is the $(i+1)$-th bit of the respective τ.

Next, we add a dependency of the used $\mathbf{k}_0$ on the $(i+1)$-th bit of τ. (That is, depending on τ_{i+1}, we will use two different values of $\mathbf{k}_0$ both for preparing challenge ciphertexts, and for answering decryption queries.) This is accomplished by adding random values $\mathbf{k}_\Delta$ with $\mathbf{k}_\Delta^\top\mathbf{A}_{\tau_{i+1}} = 0$ to $\mathbf{k}_0$. Indeed, for challenge ciphertexts, adding such $\mathbf{k}_\Delta$ values results in the same computed keys K, and thus cannot be detected. We note however that at this point, we run into a complication: since decryption queries need not have $\mathbf{c} \in \mathrm{span}(\mathbf{A}_{\tau_{i+1}})$, we cannot simply add random values $\mathbf{k}_\Delta$ with $\mathbf{k}_\Delta^\top\mathbf{A}_{\tau_{i+1}} = 0$ to $\mathbf{k}_0$. (This could be detected in case $\mathbf{c} \notin \mathrm{span}(\mathbf{A}_{\tau_{i+1}})$.) Instead, here we rely on a trick from [11], and use that even adversarial $\mathbf{c}$ values must lie in $\mathrm{span}(\mathbf{A})$ or $\mathrm{span}(\mathbf{A}_b)$ for $b \in \{0,1\}$. (This is also the reason why we will eventually have to modify and use $\mathbf{k}_1$. We give more details on this step inside.)

Once $\mathbf{k}_0$ is fully randomized, the resulting K computed upon decryption queries with $\mathbf{c} \notin \mathrm{span}(\mathbf{A})$ will also be random, and thus any such decryption query will be rejected. Hence, using the first trick above, security of our scheme follows.

We finally mention that our complete scheme generalizes to weaker assumptions, including the k-Linear family of assumptions (see Fig. 1).

Relation to existing techniques. We borrow techniques from both GHKW [9] and Hofheinz [11], but we need to modify and adapt them for our strategy in several important respects. While the argument from [9] also relies on a consistency proof that a given ciphertext lies in one of three linear subspaces ($\mathrm{span}(\mathbf{A})$ or $\mathrm{span}(\mathbf{A}_b)$), their consistency proof is very different from ours. Namely, their consistency proof is realized entirely through a combination of different *linear* hash proof systems, and requires *orthogonal* subspaces $\mathrm{span}(\mathbf{A}_b)$. This requires a large number (i.e., 2λ) of hash proof systems, and results in large public keys to accommodate their public information. Furthermore, the ciphertexts in GHKW require a larger $[\mathbf{c}] \in \mathbb{G}^{3k}$ (compared to the Kurosawa-Desmedt scheme), but no explicit proof π in C. This results in ciphertexts of the same size as ours.

On the other hand, [11] presents a scheme with an explicit consistency proof π for a statement similar to ours (and also deals with the arising technical complications sketched above similarly). But his construction and proof are aimed at a more generic setting which also accommodates the DCR assumption (both for the PKE and consistency proof constructions). As a consequence, his construction does not modify the equivalent of our secret key $\mathbf{k}_0, \mathbf{k}_1$ at all, but instead modifies ciphertexts directly. This makes larger public keys and ciphertexts with more "randomization slots" necessary (see Fig. 1), and in fact also leads to a more complicated proof. Furthermore, in the discrete-log setting, the necessary "OR"-style proofs from [11] require pairings, and thus his PKE scheme does as well. In contrast, our scheme requires only a weaker notion of "OR"-proofs, and we show how to instantiate this notion without pairings.

Crucial ingredient: efficient pairing-free OR-proofs. In the above argument, a crucial component is of course a proof π for (3). We present a designated-verifier proof π that only occupies one group element (in the DDH case) in C. While the proof nicely serves its purpose in our scheme, we also remark that our construction is not as general as one would perhaps like: in particular, honest proofs (generated with public information and a witness) can only be generated for $\mathbf{c} \in \mathrm{span}(\mathbf{A})$ (but not for $\mathbf{c} \in \mathrm{span}(\mathbf{A}_0)$ or $\mathbf{c} \in \mathrm{span}(\mathbf{A}_1)$).

Our proof system is perhaps best described as a randomized hash proof system. We will outline a slightly simpler version of the system which only proves $\mathbf{c} \in \mathrm{span}(\mathbf{A}) \vee \mathbf{c} \in \mathrm{span}(\mathbf{A}_0)$. In that scheme, the public key contains a value $[\mathbf{k}_y^\top \mathbf{A}]$, just like in a linear hash proof system (with secret key $\mathbf{k}_y$) for showing $\mathbf{c} \in \mathrm{span}(\mathbf{A})$ (see, e.g., [7]). Now given either the secret key $\mathbf{k}_y$ or a witness $\mathbf{r}$ to the fact that $\mathbf{c} = \mathbf{A}\mathbf{r}$, we can compute $[\mathbf{k}_y^\top \mathbf{c}]$. The idea of our system is to encrypt this value $[\mathbf{k}_y^\top \mathbf{c}]$ using a special encryption scheme that is parameterized over $\mathbf{c}$ (and whose public key is also part of the proof system's public key). The crucial feature of that encryption scheme is that it becomes lossy if and only if $\mathbf{c} \in \mathrm{span}(\mathbf{A}_0)$.

We briefly sketch the soundness of our proof system: we claim that even in a setting in which an adversary has access to many simulated proofs for *valid* statements (with $\mathbf{c} \in \mathrm{span}(\mathbf{A}) \cup \mathrm{span}(\mathbf{A}_0)$), it cannot forge proofs for *invalid* statements. Indeed, proofs with $\mathbf{c} \in \mathrm{span}(\mathbf{A})$ only depend on (and thus only reveal) the public key $[\mathbf{k}_y^\top \mathbf{A}]$. Moreover, by the special lossiness of our encryption scheme, proofs with $\mathbf{c} \in \mathrm{span}(\mathbf{A}_0)$ do not reveal anything about $\mathbf{k}_y$. Hence, an adversary will not gain any information about $\mathbf{k}_y$ beyond $\mathbf{k}_y^\top \mathbf{A}$. However, any valid proof for $\mathbf{c} \notin \mathrm{span}(\mathbf{A}) \cup \mathrm{span}(\mathbf{A}_0)$ would reveal the full value of $\mathbf{k}_y$, and thus cannot be forged by an adversary that sees only proofs for valid statements.

We remark that our proof system has additional nice properties, including a form of on-the-fly extensibility to more general statements (and in particular to more than two "OR branches". We formalize this type of proof systems as "qualified proof systems" inside.

Roadmap. After recalling some preliminaries in Section 2, we introduce the notion of designated-verifier proof systems in Section 3, along with an instantiation in Section 4. Finally, in Section 5, we present our encryption scheme (in form of a key encapsulation mechanism).

2 Preliminaries

2.1 Notations

We start by introducing some notation used throughout this paper. First we denote by $\lambda \in \mathbb{N}$ the security parameter. By $\mathsf{negl}\colon \mathbb{N} \to \mathbb{R}_{\geq 0}$ we denote a negligible function. For an arbitrary set $\mathcal{B}$, by $x \leftarrow_R \mathcal{B}$ we denote the process of sampling an element x from $\mathcal{B}$ uniformly at random. For any bit string $\tau \in \{0,1\}^*$, we denote by τ_i the i-th bit of τ and by $\tau_{|i} \in \{0,1\}^i$ the bit string comprising the first i bits of τ.

Let p be a prime, and $k, \ell \in \mathbb{N}$ such that $\ell > k$. Then for any matrix $\mathbf{A} \in \mathbb{Z}_p^{\ell \times k}$, we write $\overline{\mathbf{A}} \in \mathbb{Z}_p^{k \times k}$ for the upper square matrix of $\mathbf{A}$, and $\underline{\mathbf{A}} \in \mathbb{Z}_p^{(\ell-k)\times k}$ for the lower $\ell - k$ rows of $\mathbf{A}$. With

$$\mathrm{span}(\mathbf{A}) := \{\mathbf{A}\mathbf{r} \mid \mathbf{r} \in \mathbb{Z}_p^k\} \subset \mathbb{Z}_p^\ell,$$

we denote the *span* of $\mathbf{A}$.

For vectors $\mathbf{v} \in \mathbb{Z}_p^{2k}$, by $\overline{\mathbf{v}} \in \mathbb{Z}_p^k$ we denote the vector consisting of the upper k entries of $\mathbf{v}$ and accordingly by $\underline{\mathbf{v}} \in \mathbb{Z}_p^k$ we denote the vector consisting of the lower k entries of $\mathbf{v}$.

As usual by $\mathbf{A}^\top \in \mathbb{Z}_p^{k \times \ell}$ we denote the *transpose* of $\mathbf{A}$ and if $\ell = k$ and $\mathbf{A}$ is invertible by $\mathbf{A}^{-1} \in \mathbb{Z}_p^{\ell \times \ell}$ we denote the *inverse* of $\mathbf{A}$.

For $\ell \geq k$ by $\mathbf{A}^\perp$ we denote a matrix in $\mathbb{Z}_p^{\ell \times (\ell - k)}$ with $\mathbf{A}^\top \mathbf{A}^\perp = \mathbf{0}$ and rank $\ell - k$. We denote the set of all matrices with these properties as

$$\mathsf{orth}(\mathbf{A}) := \{\mathbf{A}^\perp \in \mathbb{Z}_p^{\ell \times (\ell-k)} \mid \mathbf{A}^\top \mathbf{A}^\perp = \mathbf{0} \text{ and } \mathbf{A}^\perp \text{ has rank } \ell - k\}.$$

2.2 Hash Functions

A hash function generator is a probabilistic polynomial time algorithm $\mathcal{H}$ that, on input 1^λ, outputs an efficiently computable function $\mathsf{H} : \{0,1\}^* \to \{0,1\}^\lambda$, unless domain and co-domain are explicitly specified.

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

$$\mathrm{Adv}^{\mathrm{CR}}_{\mathcal{H},\mathcal{A}}(\lambda) := \Pr\left[x \neq x' \wedge \mathsf{H}(x) = \mathsf{H}(x') \mid (x, x') \leftarrow \mathcal{A}(1^\lambda, \mathsf{H})\right] \leq \mathsf{negl}(\lambda).$$

We say a hash function is collision resistant *if it is sampled from a collision resistant hash function generator.*

Definition 2 (Universality). *We say a hash function generator $\mathcal{H}$ is* universal*, if for every $x, x' \in \{0,1\}^*$ with $x \neq x'$ it holds*

$$\Pr\left[\mathsf{h}(x) = \mathsf{h}(x') \mid \mathsf{h} \leftarrow_R \mathcal{H}(1^\lambda)\right] = \frac{1}{2^\lambda}.$$

We say a hash function is universal *if it is sampled from a universal hash function generator.*

Lemma 1 (Leftover Hash Lemma [16]). *Let $\mathcal{X}, \mathcal{Y}$ be sets, $\ell \in \mathbb{N}$ and $\mathsf{h}\colon \mathcal{X} \to \mathcal{Y}$ be a universal hash function. Then for all $X \leftarrow_R \mathcal{X}$, $U \leftarrow_R \mathcal{Y}$ and $\varepsilon > 0$ with $\log|\mathcal{X}| \geq \log|\mathcal{Y}| + 2\log\varepsilon$ we have*

$$\Delta\left((\mathsf{h}, \mathsf{h}(X)), (\mathsf{h}, U)\right) \leq \frac{1}{\varepsilon},$$

where Δ denotes the statistical distance.

2.3 Prime-Order Groups

Let **GGen** be a PPT algorithm that on input 1^λ returns a description $\mathcal{G} = (\mathbb{G}, p, P)$ of an additive cyclic group $\mathbb{G}$ of order p for a 2λ-bit prime p, whose generator is P.

We use the representation of group elements introduced in [8]. Namely, for $a \in \mathbb{Z}_p$, define $[a] = aP \in \mathbb{G}$ as the *implicit representation* of a in $\mathbb{G}$. More generally, for a matrix $\mathbf{A} = (a_{ij}) \in \mathbb{Z}_p^{\ell \times k}$ we define $[\mathbf{A}]$ as the implicit representation of $\mathbf{A}$ in $\mathbb{G}$:

$$[\mathbf{A}] := \begin{pmatrix} a_{11}P \ldots a_{1k}P \\ \\ a_{\ell 1}P \ldots a_{\ell k}P \end{pmatrix} \in \mathbb{G}^{\ell \times k}$$

Note that from $[a] \in \mathbb{G}$ it is hard to compute the value a if the discrete logarithm assumption holds in $\mathbb{G}$. Obviously, given $[a], [b] \in \mathbb{G}$ and a scalar $x \in \mathbb{Z}_p$, one can efficiently compute $[ax] \in \mathbb{G}$ and $[a+b] \in \mathbb{G}$.

We recall the definitions of the Matrix Decision Diffie-Hellman (MDDH) assumption from [8].

Definition 3 (Matrix Distribution). *Let $k, \ell \in \mathbb{N}$, with $\ell > k$ and p be a 2λ-bit prime. We call $\mathcal{D}_{\ell,k}$ a* matrix distribution *if it outputs matrices in $\mathbb{Z}_p^{\ell \times k}$ of full rank k in polynomial time.*

In the following we only consider matrix distributions $\mathcal{D}_{\ell,k}$, where for all $\mathbf{A} \leftarrow_R \mathcal{D}_{\ell,k}$ the first k rows of $\mathbf{A}$ form an invertible matrix.

The $\mathcal{D}_{\ell,k}$-Matrix Diffie-Hellman problem is, for a randomly chosen $\mathbf{A} \leftarrow_R \mathcal{D}_{\ell,k}$, to distinguish the between tuples of the form $([\mathbf{A}], [\mathbf{A}\mathbf{w}])$ and $([\mathbf{A}], [\mathbf{u}])$, where $\mathbf{w} \leftarrow_R \mathbb{Z}_p^k$ and $\mathbf{u} \leftarrow_R \mathbb{Z}_p^\ell$.

Definition 4 ($\mathcal{D}_{\ell,k}$-Matrix Diffie-Hellman $\mathcal{D}_{\ell,k}$-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 a prime order group $\mathbb{G}$ if for all PPT adversaries $\mathcal{A}$,*

$$\begin{aligned} \mathrm{Adv}^{\mathrm{mddh}}_{\mathbb{G},\mathcal{D}_{\ell,k},\mathcal{A}}(\lambda) :&= |\Pr[\mathcal{A}(\mathcal{G}, [\mathbf{A}], [\mathbf{A}\mathbf{w}]) = 1] - \Pr[\mathcal{A}(\mathcal{G}, [\mathbf{A}], [\mathbf{u}]) = 1]| \\ &\leq \mathsf{negl}(\lambda), \end{aligned}$$

*where the probabilities are taken over $\mathcal{G} := (\mathbb{G}, p, P) \leftarrow_R$ **GGen**(1^λ), $\mathbf{A} \leftarrow_R \mathcal{D}_{\ell,k}, \mathbf{w} \leftarrow_R \mathbb{Z}_p^k, \mathbf{u} \leftarrow_R \mathbb{Z}_p^\ell$.*

For $Q \in \mathbb{N}$, $\mathbf{W} \leftarrow_R \mathbb{Z}_p^{k \times Q}$ and $\mathbf{U} \leftarrow_R \mathbb{Z}_p^{\ell \times Q}$, we consider the Q-fold $\mathcal{D}_{\ell,k}$-MDDH assumption, which states that distinguishing tuples of the form $([\mathbf{A}], [\mathbf{A}\mathbf{W}])$ from $([\mathbf{A}], [\mathbf{U}])$ is hard. 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 [8] it is shown that the two problems are equivalent, where the reduction loses at most a factor $\ell - k$.

Lemma 2 (Random self-reducibility of $\mathcal{D}_{\ell,k}$-MDDH, [8]). *Let $\ell, k, Q \in \mathbb{N}$ with $\ell > k$ and $Q > \ell - k$. For any PPT adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that $T(\mathcal{B}) \approx T(\mathcal{A}) + Q \cdot \mathsf{poly}(\lambda)$ with $\mathsf{poly}(\lambda)$ independent of $T(\mathcal{A})$, and*

$$\mathrm{Adv}^{Q\text{-mddh}}_{\mathbb{G},\mathcal{D}_{\ell,k},\mathcal{A}}(\lambda) \le (\ell - k) \cdot \mathrm{Adv}^{\mathrm{mddh}}_{\mathbb{G},\mathcal{D}_{\ell,k},\mathcal{B}}(\lambda) + \frac{1}{p-1}.$$

Here

$$\mathrm{Adv}^{Q\text{-mddh}}_{\mathbb{G},\mathcal{D}_{\ell,k},\mathcal{A}}(\lambda) := \left|\Pr[\mathcal{A}(\mathcal{G}, [\mathbf{A}], [\mathbf{AW}]) = 1] - \Pr[\mathcal{A}(\mathcal{G}, [\mathbf{A}], [\mathbf{U}]) = 1]\right|,$$

where the probability is over $\mathcal{G} := (\mathbb{G}, p, P) \leftarrow_R \mathbf{GGen}(1^\lambda)$, $\mathbf{A} \leftarrow_R \mathcal{U}_{\ell,k}, \mathbf{W} \leftarrow_R \mathbb{Z}_p^{k \times Q}$ and $\mathbf{U} \leftarrow_R \mathbb{Z}_p^{\ell \times Q}$.

The uniform distribution is a particular matrix distribution that deserves special attention, as an adversary breaking the $\mathcal{U}_{\ell,k}$-MDDH assumption can also distinguish between real MDDH tuples and random tuples for all other possible matrix distributions.

Definition 5 (Uniform distribution). *Let $\ell, k \in \mathbb{N}$, with $\ell \ge k$, and a prime p. We denote by $\mathcal{U}_{\ell,k}$ the* uniform distribution *over all full-rank $\ell \times k$ matrices over $\mathbb{Z}_p$. Let $\mathcal{U}_k := \mathcal{U}_{k+1,k}$.*

Lemma 3 ($\mathcal{D}_{\ell,k}$-*MDDH* $\Rightarrow$ $\mathcal{U}_{\ell,k}$-*MDDH*, [8]). *Let $\mathcal{D}_{\ell,k}$ be a matrix distribution. For any adversary $\mathcal{A}$ on the $\mathcal{U}_{\ell,k}$-distribution, there exists an adversary $\mathcal{B}$ on the $\mathcal{D}_{\ell,k}$-assumption such that $T(\mathcal{B}) \approx T(\mathcal{A})$ and $\mathrm{Adv}^{\mathrm{mddh}}_{\mathbb{G},\mathcal{U}_{\ell,k},\mathcal{A}}(\lambda) = \mathrm{Adv}^{\mathrm{mddh}}_{\mathbb{G},\mathcal{D}_{\ell,k},\mathcal{B}}(\lambda)$.*

We state a tighter random-self reducibility property for case of the uniform distribution.

Lemma 4 (Random self-reducibility of $\mathcal{U}_{\ell,k}$-MDDH, [8]). *Let $\ell, k, Q \in \mathbb{N}$ with $\ell > k$. For any PPT adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that $T(\mathcal{B}) \approx T(\mathcal{A}) + Q \cdot \mathsf{poly}(\lambda)$ with $\mathsf{poly}(\lambda)$ independent of $T(\mathcal{A})$, and*

$$\mathrm{Adv}^{Q\text{-mddh}}_{\mathbb{G},\mathcal{U}_{\ell,k},\mathcal{A}}(\lambda) \le \mathrm{Adv}^{\mathrm{mddh}}_{\mathbb{G},\mathcal{U}_{\ell,k},\mathcal{B}}(\lambda) + \frac{1}{p-1}.$$

We also recall this property of the uniform distribution, stated in [9].

Lemma 5 ($\mathcal{U}_k$-MDDH $\Leftrightarrow$ $\mathcal{U}_{\ell,k}$-MDDH). *Let $\ell, k \in \mathbb{N}$, with $\ell > k$. For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ (and vice versa) such that $T(\mathcal{B}) \approx T(\mathcal{A})$ and $\mathrm{Adv}^{\mathrm{mddh}}_{\mathbb{G},\mathcal{U}_{\ell,k},\mathcal{A}}(\lambda) = \mathrm{Adv}^{\mathrm{mddh}}_{\mathbb{G},\mathcal{U}_k,\mathcal{B}}(\lambda)$.*

In this paper, for efficiency considerations, and to simplify the presentation of the proof systems in Section 3, we are particularly interested in the case $k = 1$, which corresponds to the DDH assumption, that we recall here.

Definition 6 (DDH). *We say that the DDH assumption holds relative to a prime order group $\mathbb{G}$ if for all PPT adversaries $\mathcal{A}$,*

$$\mathrm{Adv}^{\mathrm{ddh}}_{\mathbb{G},\mathcal{A}}(\lambda) := \left|\Pr[\mathcal{A}(\mathcal{G}, [a], [r], [ar]) = 1] - \Pr[\mathcal{A}(\mathcal{G}, [a], [r], [b]\right| \le \mathsf{negl}(\lambda),$$

where the probabilities are taken over $\mathcal{G} := (\mathbb{G}, p, P) \leftarrow_R \mathbf{GGen}(1^\lambda)$, $a, b, r \leftarrow_R \mathbb{Z}_p$.

Note that the DDH assumption is equivalent to $\mathcal{D}_{2,1}$-MDDH, where $\mathcal{D}_{2,1}$ is the distribution that outputs matrices $\binom{1}{a}$, for $a \leftarrow_R \mathbb{Z}_p$ chosen uniformly at random.

2.4 Public-Key Encryption

Definition 7 (Public-Key Encryption). *A public-key encryption scheme is a tuple of three PPT algorithms* $(\mathbf{Gen}, \mathbf{Enc}, \mathbf{Dec})$ *such that:*

$\mathbf{Gen}(1^\lambda)$*: returns a pair* (pk, sk) *of a public and a secret key.*
$\mathbf{Enc}(pk, M)$*: given a public key* pk *and a message* $M \in \mathcal{M}(\lambda)$*, returns a ciphertext* C*.*
$\mathbf{Dec}(pk, sk, C)$*: deterministically decrypts the ciphertext* C *to obtain a message* M *or a special rejection symbol* $\perp$*.*

We say $\mathbf{PKE} := (\mathbf{Gen}, \mathbf{Enc}, \mathbf{Dec})$ *is* **perfectly correct***, if for all* $\lambda \in \mathbb{N}$*,*

$$\Pr[\mathbf{Dec}(pk, sk, \mathbf{Enc}(pk, M)) = M] = 1,$$

where the probability is over $(pk, sk) \leftarrow_R \mathbf{Gen}(1^\lambda)$ *,* $C \leftarrow_R \mathbf{Enc}(pk, M)$*.*

Definition 8 (Multi-ciphertext CCA security). *For any public-key encryption scheme* $\mathbf{PKE} = (\mathbf{Gen}, \mathbf{Enc}, \mathbf{Dec})$ *and any stateful adversary* $\mathcal{A}$*, we define the following security experiment:*

$\mathrm{Exp}^{\mathrm{cca}}_{\mathbf{PKE},\mathcal{A}}(\lambda)$:	$\mathcal{O}_{\mathbf{enc}}(M_0, M_1)$:	$\mathcal{O}_{\mathbf{dec}}(C)$:
$(pk, sk) \leftarrow_R \mathbf{Gen}(1^\lambda)$	***if*** $\lvert M_0\rvert = \lvert M_1\rvert$	***if*** $C \notin \mathcal{C}_{\mathbf{enc}}$
$b \leftarrow_R \{0,1\}$	$C \leftarrow_R \mathbf{Enc}(pk, M_b)$	$M := \mathbf{Dec}(pk, sk, C)$
$\mathcal{C}_{\mathbf{enc}} := \emptyset$	$\mathcal{C}_{\mathbf{enc}} := \mathcal{C}_{\mathbf{enc}} \cup \{C\}$	***return*** M
$b' \leftarrow_R \mathcal{A}^{\mathcal{O}_{\mathbf{enc}}(\cdot,\cdot), \mathcal{O}_{\mathbf{dec}}(\cdot)}(pk)$	***return*** C	***else return*** $\perp$
if $b = b'$ ***return*** 1		
else return 0		

We say $\mathbf{PKE}$ *is IND-CCA secure, if for all PPT adversaries* $\mathcal{A}$*, the advantage*

$$\mathrm{Adv}^{\mathrm{cca}}_{\mathbf{PKE},\mathcal{A}}(\lambda) := \left| \Pr[\mathrm{Exp}^{\mathrm{cca}}_{\mathbf{PKE},\mathcal{A}}(\lambda) = 1] - \frac{1}{2} \right| \leq \mathsf{negl}(\lambda).$$

2.5 Key Encapsulation Mechanism

Instead of presenting an IND-CCA secure encryption scheme directly, we construct a key encapsulation mechanism (KEM) and prove that it satisfies the security notion of *indistinguishability against constrained chosen-ciphertext attacks* (IND-CCCA) [14]. By the results of [14], together with an arbitrary authenticated symmetric encryption scheme, this yields an IND-CCA secure hybrid encryption.[7] Roughly speaking, the CCCA security experiment, in contrast to the CCA experiment, makes an additional requirement on decryption queries. Namely, in addition to the ciphertext, the adversary has to provide a predicate implying some partial knowledge about the key to be decrypted. The idea of hybrid encryption and the notion of a KEM was first formalized in [6].

Definition 9 (Key Encapsulation Mechanism). *A* key encapsulation mechanism *is a tuple of PPT algorithms* $(\mathbf{KGen}, \mathbf{KEnc}, \mathbf{KDec})$ *such that:*

$\mathbf{KGen}(1^\lambda)$*: generates a pair* (pk, sk) *of keys.*

[7] The corresponding reduction is tight also in the multi-user and multi-ciphertext setting. Suitable (one-time) secure symmetric encryption schemes exist even unconditionally [14].

KEnc(pk): *on input* pk, *returns a ciphertext* C *and a symmetric key* $K \in \mathcal{K}(\lambda)$, *where* $\mathcal{K}(\lambda)$ *is the key-space.*

KDec(pk, sk, C): *deterministically decrypts the ciphertext* C *to obtain a key* $K \in \mathcal{K}(\lambda)$ *or a special rejection symbol bot.*

We say $(\mathbf{Gen}, \mathbf{Enc}, \mathbf{Dec})$ *is* **perfectly correct**, *if for all* $\lambda \in \mathbb{N}$,

$$\Pr[\mathbf{KDec}(pk, sk, C) = K] = 1,$$

where $(pk, sk) \leftarrow_R \mathbf{Gen}(1^\lambda)$, $(K, C) \leftarrow_R \mathbf{KEnc}(pk)$ *and the probability is taken over the random coins of* **Gen** *and* **KEnc**.

As mentioned above, for *constrained* chosen ciphertext security, the adversary has to have some knowledge about the key up front in order to make a decryption query. As in [14] we will use a measure for the uncertainty left and require it to be negligible for every query, thereby only allowing decryption queries where the adversary has a high prior knowledge of the corresponding key. We now provide a formal definition.

Definition 10 (Multi-ciphertext IND-CCCA security). *For any key encapsulation mechanism* $\mathbf{KEM} = (\mathbf{KGen}, \mathbf{KEnc}, \mathbf{KDec})$ *and any stateful adversary* $\mathcal{A}$, *we define the following experiment:*

$\mathrm{Exp}^{\mathrm{ccca}}_{\mathbf{KEM},\mathcal{A}}(\lambda)$:	$\mathcal{O}_{\mathbf{enc}}$:	$\mathcal{O}_{\mathbf{dec}}(\mathsf{pred}_i, C_i)$:
$(pk, sk) \leftarrow_R \mathbf{KGen}(1^\lambda)$	$K_0 \leftarrow_R \mathcal{K}(\lambda)$	$K_i := \mathbf{KDec}(pk, sk, C_i)$
$b \leftarrow_R \{0, 1\}$	$(C, K_1) \leftarrow_R \mathbf{KEnc}(pk)$	***if*** $C_i \notin \mathcal{C}_{\mathbf{enc}}$ ***and***
$\mathcal{C}_{\mathbf{enc}} := \emptyset$	$\mathcal{C}_{\mathbf{enc}} := \mathcal{C}_{\mathbf{enc}} \cup \{C\}$	***if*** $\mathsf{pred}_i(K_i) = 1$
$b' \leftarrow_R \mathcal{A}^{\mathcal{O}_{\mathbf{enc}}, \mathcal{O}_{\mathbf{dec}}(\cdot,\cdot)}(pk)$	***return*** (C, K_b)	***return*** K_i
if $b = b'$ ***return*** 1		***else return*** $\perp$
else return 0		

Here $\mathsf{pred}_i \colon \mathcal{K}(\lambda) \mapsto \{0, 1\}$ *denotes the predicate sent in the i-th decryption query, which is required to be provided as the description of a polynomial time algorithm (which can be enforced for instance by requiring it to be given in form of a circuit). Let further* $Q_{\mathbf{dec}}$ *be the number of total decryption queries made by* $\mathcal{A}$ *during the experiment, which are independent of the environment (hereby we refer to the environment the adversary runs in) without loss of generality. The uncertainty of knowledge about the keys corresponding to decryption queries is defined as*

$$\mathrm{uncert}_{\mathcal{A}}(\lambda) := \frac{1}{Q_{\mathbf{dec}}} \sum_{i=1}^{Q_{\mathbf{dec}}} \Pr_{K \leftarrow_R \mathcal{K}(\lambda)}[\mathsf{pred}_i(K) = 1].$$

We say that the key encapsulation mechanism **KEM** *is* IND-CCCA *secure, if for all PPT adversaries with negligible* $\mathrm{uncert}_{\mathcal{A}}(\lambda)$, *for the advantage we have*

$$\mathrm{Adv}^{\mathrm{ccca}}_{\mathbf{KEM},\mathcal{A}}(\lambda) := \left| \Pr[\mathrm{Exp}^{\mathrm{ccca}}_{\mathbf{KEM},\mathcal{A}}(\lambda) = 1] - \frac{1}{2} \right| \leq \mathsf{negl}(\lambda).$$

Note that the term $\mathrm{uncert}_{\mathcal{A}}(\lambda)$ in the final reduction (proving IND-CCA security of the hybrid encryption scheme consisting of an unconditionally one-time secure authenticated encryption scheme and an IND-CCCA secure KEM) is statistically small (due to the fact that the symmetric building block is unconditionally secure). Thus we are able obtain a tight security reduction even if the term $\mathrm{uncert}_{\mathcal{A}}(\lambda)$ is multiplied by the number of encryption and decryption queries in the security loss (as it will be the case for our construction).

3 Qualified Proof Systems

The following notion of a *proof system* is a combination of a non-interactive designated verifier proof system and a hash proof system. Our combined proofs consist of a proof Π and a key K, where the key K can be recovered by the verifier with a secret key and the proof Π. The key K can be part of the key in the key encapsulation mechanism presented later and thus will not enlarge the ciphertext size.

Definition 11 (Proof system). *Let $\mathcal{L} = \{\mathcal{L}_{pars}\}$ be a family of languages indexed by the public parameters pars, with $\mathcal{L}_{pars} \subseteq \mathcal{X}_{pars}$ and an efficiently computable witness relation $\mathcal{R}$. A* proof system *for $\mathcal{L}$ is a tuple of PPT algorithms* $(\mathbf{PGen}, \mathbf{PPrv}, \mathbf{PVer}, \mathbf{PSim})$ *such that:*

- $\mathbf{PGen}(1^\lambda)$*: generates a public key ppk and a secret key psk.*
- $\mathbf{PPrv}(ppk, x, w)$*: given a word $x \in \mathcal{L}$ and a witness w with $\mathcal{R}(x, w) = 1$, deterministically outputs a proof Π and a key K.*
- $\mathbf{PVer}(ppk, psk, x, \Pi)$*: on input ppk, psk, $x \in \mathcal{X}$ and Π, deterministically outputs a verdict $b \in \{0, 1\}$ and in case $b = 1$ additionally a key K, else $\perp$.*
- $\mathbf{PSim}(ppk, psk, x)$*: given the keys ppk, psk and a word $x \in \mathcal{X}$, deterministically outputs a proof Π and a key K.*

The following definition of a qualified proof system is a variant of "benign proof systems" as defined in [11] tailored to our purposes. Compared to benign proof systems, our proof systems feature an additional "key derivation" stage, and satisfy a weaker soundness requirement (that is of course still sufficient for our purpose). We need to weaken the soundness condition (compared to benign proof systems) in order to prove soundness of our instantiation.

We will consider soundness relative to a language $\mathcal{L}^{\text{snd}} \supseteq \mathcal{L}$. An adversary trying to break soundness has access to an oracle simulating proofs and keys for statements randomly chosen from $\mathcal{L}^{\text{snd}} \setminus \mathcal{L}$ and a verification oracle, which only replies other than $\perp$ if the adversary provides a valid proof and has a high a-priori knowledge of the corresponding key. The adversary wins if it can provide a valid verification query outside $\mathcal{L}^{\text{snd}}$. The adversary loses immediately if it provides a valid verification query in $\mathcal{L}^{\text{snd}} \setminus \mathcal{L}$. This slightly weird condition is necessitated by our concrete instantiation which we do not know how to prove sound otherwise. We will give more details in the corresponding proof in Section 4.2. The weaker notion of soundness still suffices to prove our KEM secure, because we employ soundness at a point where valid decryption queries in $\mathcal{L}^{\text{snd}} \setminus \mathcal{L}$ end the security experiment anyway.

Definition 12 (Qualified Proof System). *Let* $\mathbf{PS} = (\mathbf{PGen}, \mathbf{PPrv}, \mathbf{PVer}, \mathbf{PSim})$ *be a proof system for a family of languages $\mathcal{L} = \{\mathcal{L}_{pars}\}$. Let $\mathcal{L}^{\text{snd}} = \{\mathcal{L}^{\text{snd}}_{pars}\}$ be a family of languages, such that $\mathcal{L}_{pars} \subseteq \mathcal{L}^{\text{snd}}_{pars}$. We say that* $\mathbf{PS}$ *is* $\mathcal{L}^{\text{snd}}$-qualified*, if the following properties hold:*

Completeness: *For all possible public parameters pars, for all words $x \in \mathcal{L}$, and all witnesses w such that $\mathcal{R}(x, w) = 1$, we have*

$$\Pr[\mathbf{PVer}(ppk, psk, x, \Pi) = (1, K)] = 1,$$

where the probability is taken over $(ppk, psk) \leftarrow_R \mathbf{PGen}\,(1^\lambda)$ and $(\Pi, K) := \mathbf{PPrv}(ppk, x, w)$.

Uniqueness of the proofs: *For all possible public parameters pars, all key pairs (ppk, psk) in the output space of $\mathbf{PGen}\,(1^\lambda)$, and all words $x \in \mathcal{L}$, there exists* at most one *Π such that $\mathbf{PVer}(ppk, psk, x, \Pi)$ outputs the verdict 1.*

Perfect zero-knowledge: *For all public parameters pars, all key pairs (ppk, psk) in the range of* $\mathbf{PGen}(1^\lambda)$*, all words $x \in \mathcal{L}$, and all witnesses w with $\mathcal{R}(x, w) = 1$, we have*

$$\mathbf{PPrv}(ppk, x, w) = \mathbf{PSim}(ppk, psk, x).$$

Constrained $\mathcal{L}^{\text{snd}}$-soundness: *For any stateful PPT adversary $\mathcal{A}$, we consider the following soundness game (where* **PSim** *and* **PVer** *are implicitly assumed to have access to ppk):*

$\underline{\mathrm{Exp}^{\text{csnd}}_{\mathbf{PS},\mathcal{A}}(\lambda)}$:	$\underline{\mathcal{O}_{\mathbf{sim}}}$:	$\underline{\mathcal{O}_{\mathbf{ver}}(x, \mathit{\Pi}, \mathsf{pred})}$:

$(ppk, psk) \leftarrow_R \mathbf{PGen}(1^\lambda)$
$\mathcal{A}^{\mathcal{O}_{\mathbf{sim}}, \mathcal{O}_{\mathbf{ver}}(\cdot,\cdot,\cdot)}(1^\lambda, ppk)$
if $\mathcal{O}_{\mathbf{ver}}$ ***returned*** lose
    return 0
if $\mathcal{O}_{\mathbf{ver}}$ ***returned*** win
    return 1
return 0
	$x \leftarrow_R \mathcal{L}^{\text{snd}} \backslash \mathcal{L}$
	$(\mathit{\Pi}, K) \leftarrow \mathbf{PSim}(psk, x)$
	return $(x, \mathit{\Pi}, K)$
		$(v, K) := \mathbf{PVer}(psk, x, \mathit{\Pi})$
		if $v = 1$ ***and*** $\mathsf{pred}(K) = 1$
		    if $x \in \mathcal{L}$
		        return K
		    else if $x \in \mathcal{L}^{\text{snd}}$
		        return lose ***and abort***
		    else return win ***and abort***
		else return $\perp$

Let $Q_{\mathbf{ver}}$ be the total number of oracle queries to $\mathcal{O}_{\mathbf{ver}}$ and pred_i be the predicate submitted by $\mathcal{A}$ on the i-th query. The adversary $\mathcal{A}$ loses and the experiment aborts if the verification oracle answers lose *on some query of $\mathcal{A}$. The adversary $\mathcal{A}$ wins, if the oracle $\mathcal{O}_{\mathbf{ver}}$ returns* win *on some query $(x, \mathit{\Pi}, \mathsf{pred})$ of $\mathcal{A}$ with $x \notin \mathcal{L}^{\text{snd}}$ and the following conditions hold:*

- *The predicate corresponding to the i-th query is of the form $\mathsf{pred}_i \colon \mathcal{K} \cup \{\perp\} \to \{0, 1\}$ with $\mathsf{pred}_i(\perp) = 0$ for all $i \in \{1, \ldots, Q_{\mathbf{ver}}\}$.*
- *For all environments $\mathcal{E}$ having at most running time of the described constrained soundness experiment, we require that*

$$\mathrm{uncert}^{\text{snd}}_{\mathcal{A}}(\lambda) := \frac{1}{Q_{\mathbf{ver}}} \sum_{i=1}^{Q_{\mathbf{ver}}} \Pr_{K \in \mathcal{K}}[\mathsf{pred}_i(K) = 1 \textit{ when } \mathcal{A} \textit{ runs in } \mathcal{E}]$$

is negligible in λ.

Note that in particular the adversary cannot win anymore after the verification oracle replied lose *on one of its queries, as in this case the experiment directly aborts and outputs* 0*. Let $\mathrm{Adv}^{\text{csnd}}_{\mathcal{L}^{\text{snd}},\mathbf{PS},\mathcal{A}}(\lambda) := \Pr[\mathrm{Exp}^{\text{csnd}}_{\mathbf{PS},\mathcal{A}}(\lambda) = 1]$, where the probability is taken over the random coins of $\mathcal{A}$ and $\mathrm{Exp}^{\text{csnd}}_{\mathbf{PS},\mathcal{A}}$. Then we say* constrained $\mathcal{L}^{\text{snd}}$-soundness *holds for* **PS***, if for every PPT adversary $\mathcal{A}$, $\mathrm{Adv}^{\text{csnd}}_{\mathcal{L}^{\text{snd}},\mathbf{PS},\mathcal{A}}(\lambda) = \mathsf{negl}(\lambda)$.*

To prove security of the key encapsulation mechanism later, we need to switch between two proof systems. Intuitively this provides an additional degree of freedom, allowing to randomize the keys of the challenge ciphertexts gradually. To justify this transition, we introduce the following notion of indistinguishable proof systems.

Definition 13 ($\mathcal{L}^{\text{snd}}$-indistinguishability of two proof systems). *Let $\mathcal{L} \subseteq \mathcal{L}^{\text{snd}}$ be (families of) languages. Let $\mathbf{PS}_0 := (\mathbf{PGen}_0, \mathbf{PPrv}_0, \mathbf{PVer}_0, \mathbf{PSim}_0)$ and $\mathbf{PS}_1 := (\mathbf{PGen}_1, \mathbf{PPrv}_1, \mathbf{PVer}_1, \mathbf{PSim}_1)$ proof systems for $\mathcal{L}$. For every adversary $\mathcal{A}$, we define the following experiment (where $\mathbf{PSim}_b$ and $\mathbf{PVer}_b$ are implicitly assumed to have access to ppk):*

$\underline{\mathrm{Exp}^{\mathrm{PS-ind}}_{\mathcal{L}^{\mathrm{snd}},\mathbf{PS}_0,\mathbf{PS}_1,\mathcal{A}}(\lambda)}$:	$\underline{\mathcal{O}^b_{\mathbf{sim}}}$:	$\underline{\mathcal{O}^b_{\mathbf{ver}}(x,\Pi,\mathsf{pred})}$:
$b \leftarrow_R \{0,1\}$	$x \leftarrow_R \mathcal{L}^{\mathrm{snd}} \setminus \mathcal{L}$	$(v,K) := \mathbf{PVer}_b(psk,x,\Pi)$
$(ppk,psk) \leftarrow \mathbf{PGen}_b(1^\lambda)$	$(\Pi,K) \leftarrow \mathbf{PSim}_b(psk,x)$	***if*** $v=1$ ***and*** $\mathsf{pred}(K)=1$
$b' \leftarrow \mathcal{A}^{\mathcal{O}^b_{\mathbf{sim}},\mathcal{O}^b_{\mathbf{ver}}(\cdot,\cdot)}(ppk)$	***return*** (x,Π,K)	***and*** $x \in \mathcal{L}^{\mathrm{snd}}$
if $b=b'$ ***return*** 1		***return*** K
else return 0		***else return*** $\perp$

As soon as $\mathcal{A}$ has submitted one query which is replied with lose *by the verification oracle, the experiment aborts and outputs* 0.

We define the advantage function

$$\mathrm{Adv}^{\text{PS-ind}}_{\mathcal{L}^{\mathrm{snd}},\mathbf{PS}_0,\mathbf{PS}_1,\mathcal{A}}(\lambda) := \left| \Pr\left[\mathrm{Exp}^{\mathrm{PS-ind}}_{\mathcal{L}^{\mathrm{snd}},\mathbf{PS}_0,\mathbf{PS}_1,\mathcal{A}}(\lambda)=1\right] - \frac{1}{2} \right|.$$

*We say $\mathbf{PS}_0$ and $\mathbf{PS}_1$ are $\mathcal{L}^{\mathrm{snd}}$-*indistinguishable*, if for all (unbounded) algorithms $\mathcal{A}$ the advantage $\mathrm{Adv}^{\text{PS-ind}}_{\mathcal{L},\mathbf{PS}_0,\mathbf{PS}_1,\mathcal{A}}(\lambda)$ is negligible in λ.*

Note that we adopt a different (and simpler) definition for the verification oracle in the indistinguishability game than in the soundness game, in particular it leaks more information about the keys. We can afford this additional leakage for indistinguishability, but not for soundness.

In order to prove security of the key encapsulation mechanism presented in Section 5, we will require one proof system and the existence of a second proof system it can be extended to. We capture this property in the following definition.

Definition 14 ($\widetilde{\mathcal{L}^{\mathrm{snd}}}$-extensibility of a proof system). *Let $\mathcal{L} \subseteq \mathcal{L}^{\mathrm{snd}} \subseteq \widetilde{\mathcal{L}^{\mathrm{snd}}}$ be three (families of) languages. An $\mathcal{L}^{\mathrm{snd}}$-qualified proof system $\mathbf{PS}$ for language $\mathcal{L}$ is said to be $\widetilde{\mathcal{L}^{\mathrm{snd}}}$-*extensible *if there exists a proof system $\widetilde{\mathbf{PS}}$ for $\mathcal{L}$ that complies with $\widetilde{\mathcal{L}^{\mathrm{snd}}}$-constrained soundness and such that $\mathbf{PS}$ and $\widetilde{\mathbf{PS}}$ are $\mathcal{L}^{\mathrm{snd}}$-indistinguishable.*

4 The OR-Proof

In the following sections we explain how the public parameters $pars_{\mathbf{PS}}$ are sampled, how our system of OR-languages is defined and how to construct a qualified proof system complying with constrained soundness respective to these languages.

4.1 Public Parameters and the OR-Languages

First we need to choose a $k \in \mathbb{N}$ depending on the assumption we use to prove security of our constructions. We invoke $\mathbf{GGen}(1^\lambda)$ to obtain a group description $\mathcal{G} = (\mathbb{G},p,P)$ with $|\mathbb{G}| \geq 2^{2\lambda}$. Next we sample matrices $\mathbf{A} \leftarrow_R \mathcal{D}_{2k,k}$ and $\mathbf{A}_0 \leftarrow_R \mathcal{U}_{2k,k}$, where we assume without loss of generality that $\overline{\mathbf{A}}_0$ is full rank. Let $\mathcal{H}_0$ and $\mathcal{H}_1$ be *universal* hash function generators returning functions of the form $\mathsf{h}_0 \colon \mathbb{G}^{k+1} \to \mathbb{Z}_p^k$ and $\mathsf{h}_1 \colon \mathbb{G}^2 \to \mathbb{Z}_p$ respectively. Let $\mathsf{h}_0 \leftarrow_R \mathcal{H}_0$ and $\mathsf{h}_1 \leftarrow_R \mathcal{H}_1$.

Altogether we define the public parameters for our proof system to comprise

$$pars_{\mathbf{PS}} := (k,\mathcal{G},[\mathbf{A}],[\mathbf{A}_0],\mathsf{h}_0,\mathsf{h}_1).$$

We assume from now that all algorithms have access to $pars_{\mathbf{PS}}$ without explicitly stating it as input.

Additionally let $\mathbf{A}_1 \in \mathbb{Z}_p^{2k \times k}$ be a matrix distributed according to $\mathcal{U}_{2k,k}$ with the restriction $\overline{\mathbf{A}}_0 = \overline{\mathbf{A}}_1$. Then we define the languages

$$\begin{aligned}
\mathcal{L} &:= \operatorname{span}([\mathbf{A}]), \\
\mathcal{L}_{\text{snd}} &:= \operatorname{span}([\mathbf{A}]) \cup \operatorname{span}([\mathbf{A}_0]), \\
\widetilde{\mathcal{L}_{\text{snd}}} &:= \operatorname{span}([\mathbf{A}]) \cup \operatorname{span}([\mathbf{A}_0]) \cup \operatorname{span}([\mathbf{A}_1]).
\end{aligned}$$

A crucial building block for the key encapsulation mechanism will be a proof system $\mathbf{PS}$ that is $\mathcal{L}_{\text{snd}}$-qualified and $\widetilde{\mathcal{L}_{\text{snd}}}$-extensible. We give a construction based on $\mathcal{D}_{2k,k}$-MDDH in the following section.

4.2 The OR-Proof for $k = 1$

Our goal is to construct an $\mathcal{L}_{\text{snd}}$-qualified proof system for $\mathcal{L}$ based on $\mathcal{D}_{2k,k}$-MDDH for any matrix distribution $\mathcal{D}_{2k,k}$ (see Definition 3). To this aim we give a proof system $Pre\mathbf{PS} := (Pre\mathbf{PGen}, Pre\mathbf{PPrv}, Pre\mathbf{PVer}, Pre\mathbf{PSim})$ for $\mathcal{L}$ in Fig. 2.

In case $k = 1$ this is sufficient, namely setting $\mathbf{PGen} := Pre\mathbf{PGen}$, $\mathbf{PPrv} := Pre\mathbf{PPrv}$, $\mathbf{PVer} := Pre\mathbf{PVer}$ and $\mathbf{PSim} := Pre\mathbf{PSim}$, we can prove that $\mathbf{PS} := (\mathbf{PGen}, \mathbf{PPrv}, \mathbf{PVer}, \mathbf{PSim})$ is $\mathcal{L}_{\text{snd}}$-qualified under the DDH assumption. For the case $k > 1$ we give the construction of $\mathbf{PS}$ in Fig. 5, Section 4.4

As a compromise between generality and readability, we decided to give the proof in full detail for $k = 1$ (i.e. the DDH case), while sticking to the general matrix notation. As for $k = 1$ a vector in $\mathbb{Z}_p^k = \mathbb{Z}_p^1$ is merely a single element, we do not use bold letters to denote for instance x and r in $\mathbb{Z}_p$ (other than in Fig. 2).

$\underline{Pre\mathbf{PGen}(1^\lambda)}$:
$\mathbf{K_x} \leftarrow_R \mathbb{Z}_p^{(k+1) \times 2k}$
$\mathbf{K}_y \leftarrow_R \mathbb{Z}_p^{2 \times 2k}$
return
  $ppk := ([\mathbf{K_x A}], [\mathbf{K}_y \mathbf{A}])$
  $psk := (\mathbf{K_x}, \mathbf{K}_y)$

$\underline{Pre\mathbf{PVer}(ppk, psk, [\mathbf{c}], [\pi^\star])}$:
$\mathbf{x} := \mathsf{h}_0(\mathbf{K_x}[\mathbf{c}]) \in \mathbb{Z}_p^k$
$y := \mathsf{h}_1(\mathbf{K}_y[\mathbf{c}]) \in \mathbb{Z}_p$
$[\pi] := \overline{[\mathbf{A}_0]} \cdot \mathbf{x} + \overline{[\mathbf{c}]} \cdot y \in \mathbb{Z}_p^k$
$[\kappa] := \underline{[\mathbf{A}_0]} \cdot \mathbf{x} + \underline{[\mathbf{c}]} \cdot y \in \mathbb{Z}_p^k$
if $[\pi] = [\pi^\star]$ **return** $(1, [\kappa])$
else return $(0, \perp)$

$\underline{Pre\mathbf{PPrv}(ppk, [\mathbf{c}], \mathbf{r})}$:
$\mathbf{x} := \mathsf{h}_0([\mathbf{K_x A}]\mathbf{r}) \in \mathbb{Z}_p^k$
$y := \mathsf{h}_1([\mathbf{K}_y \mathbf{A}]\mathbf{r}) \in \mathbb{Z}_p$
return
  $[\pi] := \overline{[\mathbf{A}_0]} \cdot \mathbf{x} + \overline{[\mathbf{c}]} \cdot y$
  $[\kappa] := \underline{[\mathbf{A}_0]} \cdot \mathbf{x} + \underline{[\mathbf{c}]} \cdot y$

$\underline{Pre\mathbf{PSim}(ppk, psk, [\mathbf{c}])}$:
$\mathbf{x} := \mathsf{h}_0(\mathbf{K_x}[\mathbf{c}]) \in \mathbb{Z}_p^k$
$y := \mathsf{h}_1(\mathbf{K}_y[\mathbf{c}]) \in \mathbb{Z}_p$
return
  $[\pi] := \overline{[\mathbf{A}_0]} \cdot \mathbf{x} + \overline{[\mathbf{c}]} \cdot y$
  $[\kappa] := \underline{[\mathbf{A}_0]} \cdot \mathbf{x} + \underline{[\mathbf{c}]} \cdot y$

Fig. 2: Proof System $Pre\mathbf{PS}$ for $\mathcal{L}$. For $k = 1$ the proof system $\mathbf{PS} := Pre\mathbf{PS}$ is $\mathcal{L}_{\text{snd}}$-qualified based on DDH. For $k > 1$ we give an $\mathcal{L}_{\text{snd}}$-qualified proof system based on $\mathcal{D}_{2k,k}$-MDDH in Fig. 5 in Section 4.4.

Theorem 1. *If the DDH assumption holds in $\mathbb{G}$, and h_0, h_1 are universal hash functions, then for $k = 1$ the proof system $\mathbf{PS} := Pre\mathbf{PS}$ described in Fig. 2 is $\mathcal{L}^{\text{snd}}$-qualified. Further, the proof system $\mathbf{PS}$ is $\widetilde{\mathcal{L}_{\text{snd}}}$-extensible.*

Proof. *Completeness* and *perfect zero-knowledge* follow straightforwardly from the fact that for all $r \in \mathbb{Z}_p$, $[\mathbf{K}_x\mathbf{A}]r = \mathbf{K}_x[\mathbf{A}r]$ and $[\mathbf{K}_y\mathbf{A}]r = \mathbf{K}_y[\mathbf{A}r]$.

Uniqueness of the keys follows from the fact that the verification algorithm computes exactly one proof $[\pi]$ (plus the corresponding key $[\kappa]$), and aborts if $[\pi] \neq [\pi^\star]$.

We prove in Lemma 6 that $\mathbf{PS}$ satisfies constrained $\mathcal{L}^{\text{snd}}$-soundness.

For $\widetilde{\mathcal{L}^{\text{snd}}}$-extensibility we refer to Section 4.3. We describe a proof system $\widetilde{\mathbf{PS}}$ for $\mathcal{L}$ in Fig. 4, we prove in Lemma 7 that $\mathbf{PS}$ and $\widetilde{\mathbf{PS}}$ are $\mathcal{L}^{\text{snd}}$-indistinguishable, and in Lemma 8 that $\widetilde{\mathbf{PS}}$ complies with constrained $\widetilde{\mathcal{L}^{\text{snd}}}$-soundness. □

Lemma 6 (Constrained $\mathcal{L}^{\text{snd}}$-soundness of PS). *If the DDH assumption holds in $\mathbb{G}$, and h_0, h_1 are universal hash functions, then the proof system $\mathbf{PS}$ described in Fig. 2 (for $k = 1$) complies with constrained $\mathcal{L}^{\text{snd}}$-soundness. More precisely, for every adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that $T(\mathcal{B}) \approx T(\mathcal{A}) + (Q_{\mathbf{sim}} + Q_{\mathbf{ver}}) \cdot \mathsf{poly}(\lambda)$ and*

$$\mathrm{Adv}^{\mathrm{csnd}}_{\mathbf{PS},\mathcal{A}}(\lambda) \leq \mathrm{Adv}^{\mathrm{ddh}}_{\mathbb{G},\mathcal{B}}(\lambda) + Q_{\mathbf{ver}} \cdot \mathrm{uncert}^{\mathrm{snd}}_{\mathcal{A}}(\lambda) + (Q_{\mathbf{sim}} + Q_{\mathbf{ver}}) \cdot 2^{-\Omega(\lambda)},$$

where $Q_{\mathbf{ver}}$, $Q_{\mathbf{sim}}$ are the number of calls to $\mathcal{O}_{\mathbf{ver}}$ and $\mathcal{O}_{\mathbf{sim}}$ respectively, $\mathrm{uncert}^{\mathrm{snd}}_{\mathcal{A}}(\lambda)$ describes the uncertainty of the predicates provided by $\mathcal{A}$ (see Definition 12) and poly is a polynomial function independent of $T(\mathcal{A})$.

Note that, as explained in Section 2.5, in the proof of IND-CCA security of the final hybrid encryption scheme (where we will employ constrained $\mathcal{L}_{\text{snd}}$-soundness of $\mathbf{PS}$ to prove IND-CCCA security of our KEM), the term $\mathrm{uncert}^{\mathrm{snd}}_{\mathcal{A}}(\lambda)$ will be statistically small, so we can afford to get a security loss of $Q_{\mathbf{ver}} \cdot \mathrm{uncert}^{\mathrm{snd}}_{\mathcal{A}}(\lambda)$ without compromising tightness.

Proof. We prove $\mathcal{L}_{\text{snd}}$-*soundness* of $\mathbf{PS}$ via a series of games, described in Fig. 3. We start by giving a short overview of the proof.

The idea is to first randomize x used in simulated proofs of statements $[\mathbf{c}] \in \mathcal{L}_{\text{snd}} \setminus \mathcal{L}$, using the DDH assumption and the Leftover Hash Lemma (Lemma 1). This makes $[\pi, \kappa]$ an encryption of y that becomes lossy if and only if $[\mathbf{c}] \in \mathrm{span}([\mathbf{A}_0])$. For the final proof step, let $([\mathbf{c}], [\pi], [\kappa])$ be an honestly generated combined proof (with randomized x) with $[\mathbf{c}] \in \mathcal{L}_{\text{snd}}$, that is there exists an $r \in \mathbb{Z}_p$ such that either $[\mathbf{c}] = [\mathbf{A}r]$ or $[\mathbf{c}] = [\mathbf{A}_0 r]$. In the former case, we have $y = \mathsf{h}_1(\mathbf{K}_y^\top[\mathbf{c}]) = \mathsf{h}_1([\mathbf{K}_y\mathbf{A}]r)$, thus no information about $\mathbf{K}_y$ is leaked apart from what is already contained in the public key. In the latter case, we have $[\pi, \kappa] = [\mathbf{A}_0] \cdot x + [\mathbf{c}] \cdot y = [\mathbf{A}_0](x + r \cdot y)$, thus y, and in particular $\mathbf{K}_y$, are completely hidden by the randomized x. This implies that even knowing many sound tuples $([\mathbf{c}], [\pi], [\kappa])$ for $[\mathbf{c}] \in \mathcal{L}_{\text{snd}}$, an adversary cannot do better than guessing y to produce a valid key for a statement outside $\mathcal{L}_{\text{snd}}$, and therefore, only has negligible winning chances.

We start with the constrained $\mathcal{L}_{\text{snd}}$-soundness game, which we refer to as game $\mathbf{G}$. In the following we want to bound the probability

$$\varepsilon := \mathrm{Adv}^{\mathrm{csnd}}_{\mathbf{PS},\mathcal{A}}(\lambda).$$

We denote the probability that the adversary $\mathcal{A}$ wins the game $\mathbf{G}_i$ by

$$\varepsilon_i := \mathrm{Adv}_{\mathbf{G}_i,\mathcal{A}}(\lambda).$$

#	sim. x for $[\mathbf{c}] \in \mathcal{L}_{\text{snd}} \backslash \mathcal{L}$	ver. $[\kappa]$ for $[\mathbf{c}] \notin \mathcal{L}$	game knows	remark
$\mathbf{G}_0$	$x := \mathsf{h}_0\left(\mathbf{K}_x[\mathbf{c}]\right)$	$\underline{[\mathbf{A}_0]} \cdot \mathbf{x} + \underline{[\mathbf{c}]} \cdot y$		$\mathcal{L}_{\text{snd}}$-soundn. game w/o *lose*
$\mathbf{G}_1$	$x := \mathsf{h}_0\left(\mathbf{K}_x[\mathbf{c}]\right)$	$\underline{\mathbf{A}}_0\overline{\mathbf{A}}_0^{-1}\left([\pi^\star] - \overline{[\mathbf{c}]} \cdot y\right) + \underline{[\mathbf{c}]} \cdot y$	$\mathbf{A}, \mathbf{A}_0$	win. chances increase
$\mathbf{G}_2$	$\mathbf{u} \leftarrow_R \mathbb{Z}_p^2$, $x := \mathsf{h}_0([\mathbf{u}])$	$\underline{\mathbf{A}}_0\overline{\mathbf{A}}_0^{-1}\left([\pi^\star] - \overline{[\mathbf{c}]} \cdot y\right) + \underline{[\mathbf{c}]} \cdot y$	$\mathbf{A}, \mathbf{A}_0$	DDH
$\mathbf{G}_3$	$x \leftarrow_R \mathbb{Z}_p$	$\underline{\mathbf{A}}_0\overline{\mathbf{A}}_0^{-1}\left([\pi^\star] - \overline{[\mathbf{c}]} \cdot y\right) + \underline{[\mathbf{c}]} \cdot y$	$\mathbf{A}, \mathbf{A}_0$	Lemma 1 (LOHL)

Fig. 3: Overview of the proof of $\mathcal{L}_{\text{snd}}$-constrained soundness of **PS**. The first column shows how x is computed for queries to $\mathcal{O}_{\mathbf{sim}}$. The second column shows how the key $[\kappa]$ is computed by the verifier in queries to $\mathcal{O}_{\mathbf{ver}}$ when $[\mathbf{c}] \notin \mathcal{L}$.

$\mathbf{G} \rightsquigarrow \mathbf{G_0}$: From game $\mathbf{G}_0$ on, on a valid verification query $([\mathbf{c}], \Pi, \mathsf{pred})$ the verification oracle will not return *lose* and abort anymore, but instead simply return $\perp$. This can only increase the winning chances of an adversary $\mathcal{A}$. Thus we obtain

$$\varepsilon \leq \varepsilon_0.$$

$\mathbf{G_0} \rightsquigarrow \mathbf{G_1}$: We show that $\varepsilon_1 \geq \varepsilon_0$. The difference between $\mathbf{G}_0$ and $\mathbf{G}_1$ is that from game $\mathbf{G}_1$ on the oracle $\mathcal{O}_{\mathbf{ver}}$, on input $([\mathbf{c}], \Pi, \mathsf{pred})$, first checks if $[\mathbf{c}] \in \operatorname{span}([\mathbf{A}])$. If this is the case, $\mathcal{O}_{\mathbf{ver}}$ behaves as in game $\mathbf{G}_0$. Otherwise, it does not check if $[\pi^\star] = [\pi]$ anymore, and it computes

$$[\kappa] = \underline{\mathbf{A}}_0\overline{\mathbf{A}}_0^{-1}\left([\pi^\star] - \overline{[\mathbf{c}]} \cdot y\right) + \underline{[\mathbf{c}]} \cdot y,$$

where y is computed as in $\mathbf{G}_0$. Note that this computation requires to know $\mathbf{A}_0$, but not $\mathbf{K}_x$, since x is not computed explicitly. This will be crucial for the transition to game $\mathbf{G}_2$.

We again have to show that this can only increase the winning chances of the adversary, in particular we have to show that this change does not affect the adversaries view on non-winning queries.

First, from game $\mathbf{G}_0$ on the verification oracle $\mathcal{O}_{\mathbf{ver}}$ always returns $\perp$ on queries from $\mathcal{L}_{\text{snd}} \backslash \mathcal{L}$, and thus games $\mathbf{G}_0$ and $\mathbf{G}_1$ only differ when $\mathcal{O}_{\mathbf{ver}}$ is queried on statements with $[\mathbf{c}] \notin \mathcal{L}_{\text{snd}}$. Therefore it remains to show that for any query $([\mathbf{c}], [\pi^\star], \mathsf{pred})$ to $\mathcal{O}_{\mathbf{ver}}$ with $[\mathbf{c}] \notin \mathcal{L}_{\text{snd}}$, we have that if the query is winning in $\mathbf{G}_0$, then it is also winning in $\mathbf{G}_1$. Suppose $([\mathbf{c}], [\pi^\star], \text{pred})$ satisfies the winning condition in $\mathbf{G}_0$. Then, it must hold true that $[\pi^\star] = \overline{[\mathbf{A}_0]} \cdot \mathbf{x} + \overline{[\mathbf{c}]} \cdot y$ and $\mathsf{pred}\left(\underline{[\mathbf{A}_0]} \cdot \mathbf{x} + \underline{[\mathbf{c}]} \cdot y\right) = 1$. In $\mathbf{G}_1$, the key is computed as

$$\underline{\mathbf{A}}_0\overline{\mathbf{A}}_0^{-1}\left([\pi^\star] - \overline{[\mathbf{c}]} \cdot y\right) + \underline{[\mathbf{c}]} \cdot y = \underline{[\mathbf{A}_0]} \cdot \mathbf{x} + \underline{[\mathbf{c}]} \cdot y,$$

and thus the query is also winning in $\mathbf{G}_1$.

Note that for this step it is crucial that we only require a weakened soundness condition of our proof systems (compared to benign proof systems [11]). Namely, if instead the verification oracle in the soundness experiment $\mathcal{O}_{\mathbf{ver}}$ returned the key $[\kappa]$ for valid statements $x \in \mathcal{L}^{\text{snd}} \backslash \mathcal{L}$, we could not argue that the proof transition does necessarily at most increase the winning chances of an adversary. This holds true as in game $\mathbf{G}_1$ on a statement $x \in \mathcal{L}^{\text{snd}} \backslash \mathcal{L}$ with non-valid proof (but with valid predicate respective to the proof) the key would be returned, whereas in game $\mathbf{G}_0$ "$\perp$" would be returned.

$\mathbf{G_1} \rightsquigarrow \mathbf{G_2}$**:** In this transition, we use the DDH assumption to change the way x is computed in simulated proofs. More precisely, we build an adversary $\mathcal{B}$ such that $T(\mathcal{B}) \approx T(\mathcal{A}) + (Q_{\mathbf{ver}} + Q_{\mathbf{sim}}) \cdot \mathsf{poly}(\lambda)$ and

$$|\varepsilon_2 - \varepsilon_1| \leq \mathrm{Adv}^{\mathrm{ddh}}_{\mathbb{G},\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}.$$

Let $([\mathbf{B}], [\mathbf{h}_1, \ldots, \mathbf{h}_{Q_{\mathbf{sim}}}])$ be a $Q_{\mathbf{sim}}$-fold DDH challenge. We build the adversary $\mathcal{B}$ as follows. First $\mathcal{B}$ picks $\mathbf{A}, \mathbf{A}_0, \mathbf{A}_1$ as described in Section 4.1. Further $\mathcal{B}$ chooses $\mathbf{K}'_x \leftarrow_R \mathbb{Z}_p^{2\times 2}$ and $\mathbf{K}_y \leftarrow_R \mathbb{Z}_p^{2\times 2}$ and implicitely sets $\mathbf{K}_x = \mathbf{K}'_x + \mathbf{U}(\mathbf{A}^\perp)^\top$ for some $\mathbf{A}^\perp \in \mathsf{orth}(\mathbf{A})$, where $\mathbf{U} \in \mathbb{Z}_p^{2\times 1}$ depends on the $Q_{\mathbf{sim}}$-fold DDH challenge (and cannot be computed by $\mathcal{B}$). This will allow $\mathcal{B}$ to embed the $Q_{\mathbf{sim}}$-fold DDH challenge into simulation queries. Note that even though $\mathcal{B}$ does not know $\mathbf{K}_x$ explicitly, the special form of $\mathbf{K}_x$ still allows $\mathcal{B}$ to compute the public parameters $[\mathbf{K}_x\mathbf{A}] = [\mathbf{K}'_x\mathbf{A}]$ and $[\mathbf{K}_y\mathbf{A}]$.

For queries to $\mathcal{O}_{\mathbf{ver}}$ containing $[\mathbf{c}] \in \mathcal{L}$, in order to compute x, $\mathcal{B}$ computes $\mathbf{K}_x[\mathbf{c}] = \mathbf{K}'_x[\mathbf{c}]$ using $\mathbf{K}'_x$ (note that $\mathcal{B}$ can check if $[\mathbf{c}] \in \mathcal{L}$ since it knows $\mathbf{A}$). Answering queries to $\mathcal{O}_{\mathbf{ver}}$ for $\mathbf{c} \notin \mathcal{L}$ does not require knowledge of x. Both cases can thus be handled without concrete knowledge of $\mathbf{K}_x$.

The adversary $\mathcal{B}$ prepares for queries to the simulation oracle $\mathcal{O}_{\mathbf{sim}}$ as follows. First it chooses $w \leftarrow \mathbb{Z}_p$ and defines $[\mathbf{V}] := w \cdot [\mathbf{B}]$. Note that with overwhelming probability over the choices of $\mathbf{A}$ and $\mathbf{A}_0$, the matrix $(\mathbf{A}^\perp)^\top \mathbf{A}_0$ is full rank and thus $(\mathbf{K}'_x + \mathbf{U}(\mathbf{A}^\perp)^\top)\mathbf{A}_0$ is distributed statistically close to uniform over $\mathbb{Z}_p$. Therefore replacing $[(\mathbf{K}'_x + \mathbf{U}(\mathbf{A}^\perp)^\top)\mathbf{A}_0]$ by $[\mathbf{V}]$ is statistically indistinguishable for the adversary $\mathcal{A}$.

On the i-th query to $\mathcal{O}_{\mathbf{sim}}$, for all $i \in [Q_{\mathbf{sim}}]$, the adversary $\mathcal{B}$ defines $[\mathbf{c}_i] := \mathbf{A}_0\overline{[\mathbf{h}_i]}$ and computes $x := \mathsf{h}_0(w \cdot [\mathbf{h}_i])$. Further $\mathcal{B}$ can compute $y := \mathsf{h}_1(\mathbf{K}_y[\mathbf{c}_i])$ as before. In case of a real DDH challenge, we have $\mathbf{h}_i = \mathbf{B}r_i$ for $r_i \leftarrow_R \mathbb{Z}_p$ and thus we have $[\mathbf{c}_i] = [\mathbf{A}_0 r_i]$ and $x = \mathsf{h}_0(w \cdot [\mathbf{B}r_i]) = \mathsf{h}_0([\mathbf{V}r_i])$. By our previous considerations $[\mathbf{V}r_i]$ is statistically close to $\mathbf{K}_x[\mathbf{c}_i]$ and thus adversary $\mathcal{B}$ simulates game $\mathbf{G}_1$. In case the adversary was given a random challenge, the $\mathbf{h}_i$ are distributed uniformly at random and the adversary simulates game $\mathbf{G}_2$. Now we can employ the random self-reducibility of DDH (Lemma 2) to obtain an adversary as claimed.

Note that in order to prove this transition we require that in the definition of constrained soundness the simulation oracle returns random challenges (otherwise we would not be able to embedd the DDH challenge into simulation queries). This is another reason why we cannot directly employ the notion of benign proof systems [11].

$\mathbf{G_2} \rightsquigarrow \mathbf{G_3}$**:** As h_0 is universal, we can employ the Leftover Hash Lemma (Lemma 1) to switch $(\mathsf{h}_0, \mathsf{h}_0([\mathbf{v}]))$ to $(\mathsf{h}_0, \mathbf{u})$ in all simulation queries, where $\mathbf{u} \leftarrow_R \mathbb{Z}_p$. A hybrid argument yields

$$|\varepsilon_2 - \varepsilon_3| \leq Q_{\mathbf{sim}}/p.$$

Game $\mathbf{G_3}$**:** We show that $\varepsilon_3 \leq Q_{\mathbf{ver}} \cdot \mathrm{uncert}^{\mathrm{snd}}_{\mathcal{A}}(\lambda)$, where $Q_{\mathbf{ver}}$ is the number of queries to $\mathcal{O}_{\mathbf{ver}}$ and $\mathrm{uncert}^{\mathrm{snd}}_{\mathcal{A}}(\lambda)$ describes the uncertainty of the predicates provided by the adversary as described in Definition 12.

We use a hybrid argument over the $Q_{\mathbf{ver}}$ queries to $\mathcal{O}_{\mathbf{ver}}$. To that end, we introduce games $\mathbf{G}_{3.i}$ for $i = 0, \ldots, Q_{\mathbf{ver}}$, defined as $\mathbf{G}_3$ except that for its first i queries $\mathcal{O}_{\mathbf{ver}}$ answers $\perp$ on any query $([\mathbf{c}], [\pi], \mathsf{pred})$ with $[\mathbf{c}] \notin \mathcal{L}_{\mathrm{snd}}$. We have $\varepsilon_3 = \varepsilon_{3.0}$, $\varepsilon_{3.Q_{\mathbf{ver}}} = 0$ and we show that for all $i = 0, \ldots, Q_{\mathbf{ver}} - 1$ it holds

$$|\varepsilon_{3.i} - \varepsilon_{3.(i+1)}| \leq \Pr_{K \in \mathcal{K}}[\mathsf{pred}_{i+1}(K) = 1] + 2^{-\Omega(\lambda)},$$

where pred_{i+1} is the predicate contained in the $i+1$-th query to $\mathcal{O}_{\mathbf{ver}}$.

Games $\mathbf{G}_{3.i}$ and $\mathbf{G}_{3.(i+1)}$ behave identically on the first i queries to $\mathcal{O}_{\mathbf{ver}}$. An adversary can only distinguish between the two, if it manages to provide a valid $(i+1)$-st query $([\mathbf{c}], [\pi], \mathsf{pred})$ to $\mathcal{O}_{\mathbf{ver}}$ with $[\mathbf{c}] \notin \mathcal{L}_{\mathrm{snd}}$. In the following we bound the probability of this happening.

From queries to $\mathcal{O}_{\mathbf{sim}}$ and the first i queries to $\mathcal{O}_{\mathbf{ver}}$ the adversary can only learn valid tuples $([\mathbf{c}], [\pi], [\kappa])$ with $[\mathbf{c}] \in \mathcal{L}_{\mathrm{snd}}$. As explained in the beginning, such combined proofs reveal nothing about $\mathbf{K}_y$ beyond what is already revealed in the public key, as either $[\mathbf{c}] = [\mathbf{A}r]$ for an $r \in \mathbb{Z}_p$ and $y = \mathsf{h}_1([\mathbf{K}_y\mathbf{c}]) = \mathsf{h}_1([\mathbf{K}_y\mathbf{A}]r)$ or $[\mathbf{c}] = [\mathbf{A}_0 r]$ and $[\pi, \kappa] = [\mathbf{A}_0](x + r \cdot y)$. In the former case y itself reveals no more about $\mathbf{K}_y$ than the public key, while in the latter case y is hidden by the fully randomized x.

For any $[\mathbf{c}] \notin \mathcal{L}_{\mathrm{snd}}$, $y = \mathsf{h}_1[\mathbf{K}_y\mathbf{c}]$ computed by $\mathcal{O}_{\mathbf{ver}}$ is distributed statistically close to uniform from the adversary's point of view because of the following. First we can replace $\mathbf{K}_y$ by $\mathbf{K}_y + \mathbf{U}(\mathbf{A}^{\perp})^{\top}$ for $\mathbf{U} \leftarrow_R \mathbb{Z}_p^{2\times 1}$ and $\mathbf{A}^{\perp} \in \mathsf{orth}(\mathbf{A})$ as both are distributed identically. By our considerations, this extra term is neither revealed through the public key, nor through the previous queries to $\mathcal{O}_{\mathbf{sim}}$ and $\mathcal{O}_{\mathbf{ver}}$.

Now Lemma 1 (Leftover Hash Lemma) implies that the distribution of y is statistically close to uniform as desired. Since $[\mathbf{c}] \notin \mathrm{span}([\mathbf{A}_0])$ we have $\underline{[\mathbf{c}]} - \underline{[\mathbf{A}_0]}\overline{\mathbf{A}}_0^{-1}\overline{[\mathbf{c}]} \neq 0$, thus the key

$$[\kappa] := \underline{\mathbf{A}}_0\overline{\mathbf{A}}_0^{-1}[\pi^\star] + \underbrace{\left(\underline{[\mathbf{c}]} - \underline{\mathbf{A}}_0\overline{\mathbf{A}}_0^{-1}\overline{[\mathbf{c}]}\right)}_{\neq 0} \cdot y$$

computed by $\mathcal{O}_{\mathbf{ver}}$ is statistically close to uniform over $\mathbb{Z}_p$. Altogether we obtain:

$$\varepsilon_3 \leq Q_{\mathbf{ver}} \cdot \mathrm{uncert}_{\mathcal{A}}^{\mathrm{snd}}(\lambda) + Q_{\mathbf{ver}} \cdot 2^{-\Omega(\lambda)}.$$

□

4.3 Extensibility to a Three-Way OR-Proof

In the following we prove that the proof system in Fig. 2 (respectively in Fig. 5 for $k > 1$) satisfies $\mathcal{L}_{\mathrm{snd}}$-extensibility (see Definition 14). Let $\mathbf{A}_1$ be defined as in section Section 4.1. In this section we implicitly assume all algorithms to have access to $pars_{\widetilde{\mathbf{PS}}} := (pars'_{\mathbf{PS}}, [\mathbf{A}_1])$.

We describe a proof system $\widetilde{Pre\mathbf{PS}}$ for $\mathcal{L}$ in Fig. 4. In case $k = 1$, $\widetilde{\mathbf{PS}} := \widetilde{Pre\mathbf{PS}}$ fulfills the requirements of $\mathcal{L}_{\mathrm{snd}}$-extensibility. We prove that it is $\mathcal{L}_{\mathrm{snd}}$-indistinguishable to $\mathbf{PS}$ in Lemma 7, and prove that it complies with constrained $\widetilde{\mathcal{L}_{\mathrm{snd}}}$-soundess in Lemma 8. For case $k > 1$ we provide the proof system $\widetilde{\mathbf{PS}}$ in Fig. 15, in Appendix A.

Lemma 7 ($\mathcal{L}_{\mathrm{snd}}$-indistinguishability). *For $k = 1$ the proof systems $\mathbf{PS}$ and $\widetilde{\mathbf{PS}}$ described in Fig. 2 and Fig. 4, resp., are $\mathcal{L}_{\mathrm{snd}}$-indistinguishable. That is, for every (unbounded) adversary $\mathcal{A}$ we have* $\mathrm{Adv}^{\text{PS-ind}}_{\mathcal{L}_{\mathrm{snd}},\mathbf{PS},\widetilde{\mathbf{PS}},\mathcal{A}}(\lambda) = 2^{-\Omega(\lambda)}$.

$\widetilde{Pre\mathbf{PGen}}(1^\lambda)$:

$\mathbf{K}_{\mathbf{x}} \leftarrow_R \mathbb{Z}_p^{(k+1)\times 2k}$
$\mathbf{K}_y, \widetilde{\mathbf{K}}_y \leftarrow_R \mathbb{Z}_p^{2\times 2k}$
$\mathbf{A}^\perp \in \mathsf{orth}(\mathbf{A})$
return
  $ppk := ([\mathbf{K}_{\mathbf{x}}\mathbf{A}], [\mathbf{K}_y\mathbf{A}])$
  $psk := (\mathbf{K}_{\mathbf{x}}, \mathbf{K}_y, \widetilde{\mathbf{K}}_y, \mathbf{A}^\perp)$

$\widetilde{Pre\mathbf{PVer}}(ppk, psk, [\mathbf{c}], [\pi^\star])$:

$\mathbf{x} := \mathsf{h}_0(\mathbf{K}_{\mathbf{x}}[\mathbf{c}]) \in \mathbb{Z}_p^k$
if $[\mathbf{c}]^\top \mathbf{A}^\perp = [\mathbf{0}]$
  $y := \mathsf{h}_1(\mathbf{K}_y[\mathbf{c}]) \in \mathbb{Z}_p$
else
  $y := \mathsf{h}_1(\widetilde{\mathbf{K}}_y[\mathbf{c}]) \in \mathbb{Z}_p$
$[\pi] := \overline{[\mathbf{A}_0]} \cdot \mathbf{x} + \overline{[\mathbf{c}]} \cdot y$
$[\kappa] := \underline{[\mathbf{A}_0]} \cdot \mathbf{x} + \underline{[\mathbf{c}]} \cdot y$
if $[\pi] = [\pi^\star]$ **return** $(1, [\kappa])$
else return $(0, \perp)$

$\widetilde{Pre\mathbf{PPrv}}(ppk, [\mathbf{c}], \mathbf{r})$:

$\mathbf{x} := \mathsf{h}_0([\mathbf{K}_{\mathbf{x}}\mathbf{A}]\mathbf{r}) \in \mathbb{Z}_p^k$
$y := \mathsf{h}_1([\mathbf{K}_y\mathbf{A}]\mathbf{r}) \in \mathbb{Z}_p$
return
  $[\pi] := \overline{[\mathbf{A}_0]} \cdot \mathbf{x} + \overline{[\mathbf{c}]} \cdot y$
  $[\kappa] := \underline{[\mathbf{A}_0]} \cdot \mathbf{x} + \underline{[\mathbf{c}]} \cdot y$

$\widetilde{Pre\mathbf{PSim}}(ppk, psk, [\mathbf{c}])$:

$\mathbf{x} := \mathsf{h}_0(\mathbf{K}_{\mathbf{x}}[\mathbf{c}]) \in \mathbb{Z}_p^k$
if $[\mathbf{c}]^\top \mathbf{A}^\perp = [\mathbf{0}]$
  $y := \mathsf{h}_1(\mathbf{K}_y[\mathbf{c}]) \in \mathbb{Z}_p$
else
  $y := \mathsf{h}_1(\widetilde{\mathbf{K}}_y[\mathbf{c}]) \in \mathbb{Z}_p$
return
  $[\pi] := \overline{[\mathbf{A}_0]} \cdot \mathbf{x} + \overline{[\mathbf{c}]} \cdot y$
  $[\kappa] := \underline{[\mathbf{A}_0]} \cdot \mathbf{x} + \underline{[\mathbf{c}]} \cdot y$

Fig. 4: Proof System $\widetilde{Pre\mathbf{PS}}$ for $\mathcal{L}$. For $k = 1$ the proof system $\widetilde{\mathbf{PS}} := \widetilde{Pre\mathbf{PS}}$ is $\widetilde{\mathcal{L}_{\text{snd}}}$-qualified based on DDH. For $k > 1$ we give a proof system whose constrained $\widetilde{\mathcal{L}^{\text{snd}}}$-soundness is based on $\mathcal{D}_{2k,k}$-MDDH in Fig. 15.

Proof. $\mathbf{PS}$ only differs from $\widetilde{\mathbf{PS}}$ for statements $[\mathbf{c}] \notin \mathcal{L}$, and since we are interested in $\mathcal{L}_{\text{snd}}$-indistinguishability, it suffices to consider $[\mathbf{c}] \in \text{span}([\mathbf{A}_0])$. To argue that the two proof systems are statistically indistinguishable for statements $[\mathbf{c}] \in \text{span}([\mathbf{A}_0])$, we use the following.

First, $\mathbf{K}_y$ and $\mathbf{K}_y + \mathbf{U}(\mathbf{A}^\perp)^\top$ are identically distributed for $\mathbf{K}_y \leftarrow_R \mathbb{Z}_p^{2\times 2}$, $\mathbf{U} \leftarrow_R \mathbb{Z}_p^{2\times 1}$, and $\mathbf{A}^\perp \in \mathsf{orth}(\mathbf{A})$. Note that the extra term $\mathbf{U}(\mathbf{A}^\perp)^\top$ does not show up in either the pk or in the oracle of the $\mathcal{L}_{\text{snd}}$-indistinguishability game for statements $[\mathbf{c}] \in \text{span}([\mathbf{A}])$ since for all $\mathbf{c} \in \text{span}(\mathbf{A})$ we have $(\mathbf{K}_y + \mathbf{U}(\mathbf{A}^\perp)^\top)\mathbf{c} = \mathbf{K}_y\mathbf{c}$.

Further for all $\mathbf{c} \in \text{span}(\mathbf{A}_0)$, $\mathbf{A}_0^\perp \in \mathsf{orth}(\mathbf{A}_0)$, we have

$$\mathbf{U}(\mathbf{A}^\perp)^\top \mathbf{c} = \left(\mathbf{U}(\mathbf{A}^\perp)^\top + \mathbf{U}_0(\mathbf{A}_0^\perp)^\top\right)\mathbf{c},$$

where $\mathbf{U}, \mathbf{U}_0 \leftarrow_R \mathbb{Z}_p^{2\times 1}$.

With probability $1 - 2^{-\Omega(\lambda)}$ over the choices of $\mathbf{A}, \mathbf{A}_0$ the vectors $\mathbf{A}^\perp$ and $\mathbf{A}_0^\perp$ together form a basis of $\mathbb{Z}_p^2$, in which case the matrix $\mathbf{U}(\mathbf{A}^\perp)^\top + \mathbf{U}_0(\mathbf{A}_0^\perp)^\top$ is distributed uniformly random over $\mathbb{Z}_p^{2\times 2}$.

In conclusion, with overwhelming probability over the choice of the public parameters we obtain that for all $\mathbf{c} \in \text{span}(\mathbf{A}_0)$, $(\mathbf{K}_y\mathbf{A}, \mathbf{K}_y\mathbf{c})$ is identically distributed to $(\mathbf{K}_y\mathbf{A}, \widetilde{\mathbf{K}}_y\mathbf{c})$, where $\widetilde{\mathbf{K}}_y \leftarrow_R \mathbb{Z}_p^{2\times 2}$ is chosen uniformly at random, independently of $\mathbf{K}_y$. This proves the lemma. ∎

$\underline{\mathbf{PGen}(1^\lambda)}$:
$(ppk_1, psk_1) \leftarrow Pre\mathbf{PGen}(1^\lambda)$
$(ppk_2, psk_2) \leftarrow Pre\mathbf{PGen}(1^\lambda)$
return
$ppk := (ppk_1, ppk_2)$
$psk := (psk_1, psk_2)$

$\underline{\mathbf{PVer}(ppk, psk, [\mathbf{c}], [\pi^\star])}$:
$[\pi_1, \kappa_1] := Pre\mathbf{PSim}(ppk_1, psk_1, [\mathbf{c}])$
$[\pi_2, \kappa_2] := Pre\mathbf{PSim}(ppk_2, psk_2, [\mathbf{c}])$
if $[\pi_1, \pi_2] = [\pi^\star]$
return $(1, [\mathsf{h}_2([\kappa_1, \kappa_2])])$
else return $(0, \perp)$

$\underline{\mathbf{PPrv}(ppk, [\mathbf{c}], \mathbf{r})}$:
$[\pi_1, \kappa_1] := Pre\mathbf{PPrv}(ppk_1, [\mathbf{c}], \mathbf{r})$
$[\pi_2, \kappa_2] := Pre\mathbf{PPrv}(ppk_2, [\mathbf{c}], \mathbf{r})$
return
$[\pi] := [\pi_1, \pi_2]$
$[\kappa] := [\mathsf{h}_2([\kappa_1, \kappa_2])]$

$\underline{\mathbf{PSim}(ppk, psk, [\mathbf{c}])}$:
$[\pi_1, \kappa_1] := Pre\mathbf{PSim}(ppk_1, psk_1, [\mathbf{c}])$
$[\pi_2, \kappa_2] := Pre\mathbf{PSim}(ppk_2, psk_2, [\mathbf{c}])$
return
$[\pi] := [\pi_1, \pi_2]$
$[\kappa] := [\mathsf{h}_2([\kappa_1, \kappa_2])]$

Fig. 5: $\mathcal{L}_{\text{snd}}$-qualified Proof System $\mathbf{PS}$ for $\mathcal{L}$ in case $k > 1$, where $Pre\mathbf{PS}$ is defined as in Fig. 2.

As the techniques used for the proof of constrained $\widetilde{\mathcal{L}_{\text{snd}}}$-soundness of $\widetilde{\mathbf{PS}}$ are very similar to the ones presented in the proof of Lemma 6 we refer to Appendix A for the details.

Lemma 8 (Constrained $\widetilde{\mathcal{L}_{\text{snd}}}$-soundness of $\widetilde{\mathbf{PS}}$ for $k = 1$). *If the DDH assumption holds in $\mathbb{G}$ and h_0, h_1 are universal hash functions, then the proof system described in Fig. 4 complies with constrained $\widetilde{\mathcal{L}_{\text{snd}}}$-soundness. Namely, for any adversary $\mathcal{A}$ against $\widetilde{\mathcal{L}_{\text{snd}}}$-soundness, there exists an adversary $\mathcal{B}$ such that $T(\mathcal{B}) \approx T(\mathcal{A}) + (Q_{\mathbf{dec}} + Q_{\mathbf{ver}}) \cdot \mathsf{poly}(\lambda)$ and*

$$\begin{aligned}\mathrm{Adv}^{\mathrm{csnd}}_{\widetilde{\mathcal{L}_{\text{snd}}}, \widetilde{\mathbf{PS}}, \mathcal{A}}(\lambda) \leq \mathrm{Adv}^{\mathrm{mddh}}_{\mathbb{G}, \mathcal{B}, \mathcal{D}_{2k,k}}(\lambda) + Q_{\mathbf{ver}} \cdot \mathrm{uncert}_{\mathcal{A}}(\lambda) \\ + (Q_{\mathbf{sim}} + Q_{\mathbf{ver}}) \cdot 2^{-\Omega(\lambda)},\end{aligned}$$

where $Q_{\mathbf{ver}}$, $Q_{\mathbf{dec}}$ are the number of calls to $\mathcal{O}_{\mathbf{ver}}$ and $\mathcal{O}_{\mathbf{dec}}$ respectively, $\mathrm{uncert}^{\mathrm{snd}}_{\mathcal{A}}(\lambda)$ describes the uncertainty of the predicates provided by $\mathcal{A}$ and poly is a polynomial function, independent of $T(\mathcal{A})$.

4.4 The OR-Proof for $k > 1$

The obstacle for $k > 1$ is that a value $y \in \mathbb{Z}_p$ cannot fully randomize a key $[\kappa]$ of dimension k. To overcome this we employ another universal hash function on $[\kappa]$. In order to obtain enough entropy, we basically have to double the basic hash proof system. Let $Pre\mathbf{PS} := (Pre\mathbf{PGen}, Pre\mathbf{PPrv}, Pre\mathbf{PVer}, Pre\mathbf{PSim})$ be the basic proof system as given in Fig. 2. Recall the public parameters $pars_{\mathbf{PS}} = (k, \mathcal{G}, [\mathbf{A}], [\mathbf{A}_0], \mathsf{h}_0, \mathsf{h}_1)$ (as defined in Section 4.1). Let further $\mathcal{H}_2$ a universal hash function generator returning functions of the form $\mathsf{h}_2\colon \mathbb{G}^{2k} \to \mathbb{Z}_p$ and let $\mathsf{h}_2 \leftarrow_R \mathcal{H}_2$. In this section we implicitly assume all algorithms to have access to $pars'_{\mathbf{PS}} = (pars_{\mathbf{PS}}, \mathsf{h}_2)$.

The proof system $\mathbf{PS}$ for $k > 1$ can be found in Fig. 5. In Theorem 2 we state the qualified soundness and extensibility of $\mathbf{PS}$. For a proof we refer to Appendix A.

Theorem 2. *If the $\mathcal{D}_{2k,k}$-MDDH assumption holds in $\mathbb{G}$ and h_0, h_1, and h_2 are universal hash functions, then the proof system $\mathbf{PS}$ described in Fig. 5 is $\mathcal{L}_{\text{snd}}$-qualified. Further, the proof system $\mathbf{PS}$ is $\widetilde{\mathcal{L}_{\text{snd}}}$-extensible.*

5 Key Encapsulation Mechanism

In this section we present our CCCA-secure KEM that builds upon a qualified proof system for the OR-language as presented in Section 4.

Ingredients. Let $pars_{\mathbf{PS}}$ be the public parameters for the underlying qualified proof system comprising $\mathcal{G} = (\mathbb{G}, p, P)$ and $\mathbf{A}, \mathbf{A}_0 \in \mathbb{Z}_p^{2k \times k}$ (as defined in Section 4.1). Recall that $\mathcal{L} = \mathrm{span}([\mathbf{A}])$, $\mathcal{L}_{\mathrm{snd}} = \mathrm{span}([\mathbf{A}]) \cup \mathrm{span}([\mathbf{A}_0])$ and $\widetilde{\mathcal{L}_{\mathrm{snd}}} = \mathrm{span}([\mathbf{A}]) \cup \mathrm{span}([\mathbf{A}_0]) \cup \mathrm{span}([\mathbf{A}_1])$ (for $\mathbf{A}_1 \in \mathbb{Z}_p^{2k \times k}$ as in Section 4.1). Let further $\mathcal{H}$ be a collosion resistant hash function generator returning functions of the form $\mathsf{H} \colon \mathbb{G}^k \to \{0,1\}^\lambda$ and let $\mathsf{H} \leftarrow_R \mathcal{H}$. We will sometimes interpret values $\tau \in \{0,1\}^\lambda$ in the image of H as elements in $\mathbb{Z}_p$ via the map $\tau \mapsto \sum_{i=1}^{\lambda} \tau_i \cdot 2^{i-1}$.

In the following we assume that all algorithms implicitly have access to the public parameters $pars_{\mathbf{KEM}} := (pars_{\mathbf{PS}}, \mathsf{H})$.

Proof systems. We employ an $\mathcal{L}_{\mathrm{snd}}$-qualified and $\widetilde{\mathcal{L}_{\mathrm{snd}}}$-extensible proof system $\mathbf{PS} := (\mathbf{PGen}, \mathbf{PPrv}, \mathbf{PVer}, \mathbf{PSim})$ for the language $\mathcal{L}$ as provided in Fig. 2 (respectively for $k > 1$ as provided in Fig. 5). We additionally require that the key space is a subset of $\mathbb{G}$, which is satisfied by our construction in Section 4.

Construction. The construction of the KEM is given in Fig. 6.

$\underline{\mathbf{KGen}(1^\lambda)}$:	$\underline{\mathbf{KEnc}(pk)}$:
$(ppk, psk) \leftarrow_R \mathbf{PGen}(1^\lambda)$	$\mathbf{r} \leftarrow_R \mathbb{Z}_p^k$
$\mathbf{k}_0, \mathbf{k}_1 \leftarrow_R \mathbb{Z}_p^{2k}$	$[\mathbf{c}] := [\mathbf{A}]\mathbf{r}$
return	$(\Pi, [\kappa]) := \mathbf{PPrv}(ppk, [\mathbf{c}], \mathbf{r})$
$\quad pk := (ppk, [\mathbf{k}_0^\top \mathbf{A}], [\mathbf{k}_1^\top \mathbf{A}])$	$\tau := \mathsf{H}(\overline{[\mathbf{c}]})$
$\quad sk := (psk, \mathbf{k}_0, \mathbf{k}_1)$	**return**
	$\quad C := ([\mathbf{c}], \Pi)$
	$\quad K := ([\mathbf{k}_0^\top \mathbf{A}] + \tau [\mathbf{k}_1^\top \mathbf{A}])\mathbf{r} + [\kappa]$
	$\underline{\mathbf{KDec}(pk, sk, C)}$:
	parse $C := ([\mathbf{c}], \Pi)$
	$(b, [\kappa]) := \mathbf{PVer}(psk, [\mathbf{c}], \Pi)$
	if $b = 0$ **return** $\perp$
	$\tau := \mathsf{H}(\overline{[\mathbf{c}]})$
	return $K := (\mathbf{k}_0 + \tau \mathbf{k}_1)^\top [\mathbf{c}] + [\kappa]$

Fig. 6: Construction of the KEM

Efficiency. When using our qualified proof system from Section 4 (respectively for $k > 1$ from Section 4.4) to instantiate $\mathbf{PS}$, the public parameters comprise $4k^2$ group elements (plus the descriptions of the group itself and four hash functions). Further public keys and ciphertexts of our KEM contain $8k + 2k^2$, resp. $4k$ group elements for $k > 1$.

We stress that our scheme does not require pairings and can be implemented with $k = 1$, resulting in a tight security reduction to the DDH assumption in $\mathbb{G}$. As in this case the upper entries of the matrix $\mathbf{A}$ is 1, we get by with 3 group elements in the public parameters. Further, we can save one hash function due to the simpler underlying proof system. For the same reason, in case $k=1$ public keys and ciphertexts contain 6, resp. 3 group elements. Compared to the GHKW scheme [9], our scheme thus has ciphertexts of the same size, but significantly smaller public keys.

Without any optimizations, encryption and decryption take $8k^2+12k$, resp. $6k^2+14k$ exponentiations for $k>1$. For DDH we have 11 for both cases (again due to the simpler proof system and the distribution). Since most of these are multi-exponentiations, however, there is room for optimizations. In comparison, encryption and decyption in the GHKW scheme take $3k^2+k$, resp. $3k$ exponentiations (plus about λk group operations for encryption, and again with room for optimizations). The main reason for our somewhat less efficient operations is the used qualified proof system. We explicitly leave open the construction of a more efficient proof system.

To turn the KEM into a IND-CCA secure hybrid encryption scheme, we require a quantitatively stronger security of the symmetric building block than [9]. Namely, the uncertainty $\mathrm{uncert}_{\mathcal{A}}(\lambda)$ in our scheme has a stronger dependency on the number of queries ($Q_{\mathbf{enc}} \cdot Q_{\mathbf{dec}}$ instead of $Q_{\mathbf{enc}}+Q_{\mathbf{dec}}$). This necessitates to increase the key size of the authenticated encryption scheme compared to [9]. Note though that one-time secure authenticated encryption schemes even exist unconditionally and therefore in the reduction proving security of the hybrid encryption scheme, the uncertainty $\mathrm{uncert}_{\mathcal{A}}(\lambda)$ will be statistically small.

Theorem 3 (Security of the KEM). *If* $\mathbf{PS}$ *is* $\mathcal{L}_{\mathrm{snd}}$*-qualified and* $\widetilde{\mathcal{L}_{\mathrm{snd}}}$*-extensible to* $\widetilde{\mathbf{PS}}$*, if* H *is a collision resistant hash function and if the* $\mathcal{D}_{2k,k}$*-MDDH assumption holds in* $\mathbb{G}$*, then the key encapsulation mechanism* **KEM** *described in Fig. 6 is perfectly correct and IND-CCCA secure. More precisely, for every IND-CCCA adversary* $\mathcal{A}$ *that makes at most* $Q_{\mathbf{enc}}$ *encryption and* $Q_{\mathbf{dec}}$ *decryption queries, there exist adversaries* $\mathcal{B}^{\mathrm{mddh}}$, $\mathcal{B}^{\mathrm{csnd}}$, $\mathcal{B}^{\mathrm{ind}}$, $\mathcal{B}^{\widetilde{\mathrm{csnd}}}$ *and* $\mathcal{B}^{\mathrm{cr}}$ *with running time* $T(\mathcal{B}^{\mathrm{mddh}}) \approx T(\mathcal{B}^{\mathrm{csnd}}) \approx T(\mathcal{B}^{\mathrm{ind}}) \approx T(\mathcal{B}^{\mathrm{csnd}}) \approx T(\mathcal{B}^{\mathrm{cr}}) \approx T(\mathcal{A}) + (Q_{\mathbf{enc}}+Q_{\mathbf{dec}}) \cdot \mathsf{poly}(\lambda)$ *respectively* $T(\mathcal{B}^{\widetilde{\mathrm{csnd}}}) \approx T(\mathcal{A}) + (Q_{\mathbf{enc}} + Q_{\mathbf{enc}} \cdot Q_{\mathbf{dec}}) \cdot \mathsf{poly}(\lambda)$ *where* poly *is a polynomial independent of* $T(\mathcal{A})$*, and such that*

$$
\begin{aligned}
\mathrm{Adv}^{\mathrm{ccca}}_{\mathbf{KEM},\mathcal{A}}(\lambda) \leq{}& \frac{1}{2} \cdot \mathrm{Adv}^{\mathrm{csnd}}_{\mathcal{L}_{\mathrm{snd}},\mathbf{PS},\mathcal{B}^{\mathrm{csnd}}}(\lambda) + \frac{1}{2} \cdot \mathrm{Adv}^{\mathrm{ind}}_{\mathcal{L}_{\mathrm{snd}},\mathbf{PS},\widetilde{\mathbf{PS}},\mathcal{B}^{\mathrm{ind}}}(\lambda) \\
&+ (2\lambda+2+k) \cdot \mathrm{Adv}^{\mathrm{mddh}}_{\mathbb{G},\mathcal{D}_{2k,k},\mathcal{B}^{\mathrm{mddh}}}(\lambda) \\
&+ \frac{\lambda}{2} \cdot \mathrm{Adv}^{\mathrm{csnd}}_{\widetilde{\mathcal{L}_{\mathrm{snd}}},\widetilde{\mathbf{PS}},\mathcal{B}^{\widetilde{\mathrm{csnd}}}}(\lambda) \\
&+ \frac{\lambda+2}{2} \cdot Q_{\mathbf{enc}} \cdot Q_{\mathbf{dec}} \cdot \mathrm{uncert}_{\mathcal{A}}(\lambda) \\
&+ \mathrm{Adv}^{\mathrm{cr}}_{\mathsf{H},\mathcal{B}^{\mathrm{cr}}}(\lambda) + Q_{\mathbf{enc}} \cdot 2^{-\Omega(\lambda)}.
\end{aligned}
$$

Proof. We use a series of games to prove the claim. We denote the probability that the adversary $\mathcal{A}$ wins the i-th Game $\mathbf{G}_i$ by ε_i. An overview of all games is given in Fig. 7.

The goal is to randomize the keys of all challenge ciphertexts and thereby reducing the advantage of the adversary to 0. The methods employed here for a tight security reduction require us to ensure that $\mathcal{O}_{\mathbf{dec}}$ aborts on ciphertexts which are not in the span of $[\mathbf{A}]$, as we will no longer be able to answer those. The justification of this step relies crucially on the additional consistency proof Π and is outsourced in Lemma 9.

Game $\mathbf{G_0}$: This game is the IND-CCCA security game (Definition 10).

#	ch. c	ch. $[\kappa]$	$\mathcal{O}_{\text{dec}}$ checks	remark
$\mathbf{G}_0$	$\mathbf{A}$	**PPrv**		IND-CCCA
$\mathbf{G}_1$	$\mathbf{A}$	**PPrv**	τ fresh	coll. resist. of H
$\mathbf{G}_2$	$\mathbf{A}$	**PSim**	τ fresh	ZK of **PS**
$\mathbf{G}_3$	$\mathbf{A}_0$	**PSim**	τ fresh	$\mathcal{D}_{2k,k}$-MDDH
$\mathbf{G}_4$	$\mathbf{A}_0$	**PSim**	τ fresh, $[\mathbf{c}] \in \text{span}([\mathbf{A}])$	Lemma 9
$\mathbf{G}_5$	$\mathbf{A}_0$	rand	τ fresh, $[\mathbf{c}] \in \text{span}([\mathbf{A}])$	$\mathcal{D}_{2k,k}$-MDDH

Fig. 7: Security of the KEM. Here column "**ch. c**" refers to the vector computed by $\mathcal{O}_{\mathbf{enc}}$ as part of the challenge ciphertexts, where $\mathbf{A}$ indicates that $[\mathbf{c}] \leftarrow_R \text{span}([\mathbf{A}])$, for instance. Column "**ch.** $[\kappa]$" refers to the key computed by $\mathcal{O}_{\mathbf{enc}}$ as part of the key K. In the column "$\mathcal{O}_{\text{dec}}$ **checks**" we describe what $\mathcal{O}_{\text{dec}}$ checks on input $C = (\mathsf{pred}, ([\mathbf{c}], \Pi))$ additionally to $C \notin \mathcal{C}_{\text{enc}}$ and $\text{pred}(K) = 1$. By a *fresh* tag $\tau := \mathsf{H}(\overline{[\mathbf{c}]})$ we denote a tag not previously used in any encryption query. In case the check fails, the decryption oracle outputs $\perp$.

$\mathbf{G_0} \rightsquigarrow \mathbf{G_1}$**:** From game $\mathbf{G}_1$ on, we restrict the adversary to decryption queries with a fresh tag, that is, a tag which has not shown up in any previous encryption query. There are two conceivable bad events, where the adversary reuses a tag.

The first event is due to a collision of the hash function. That is, $\mathcal{A}$ provides a decryption query $([\mathbf{c}], \Pi)$, such that there exists a challenge ciphertext $[\mathbf{c}']$ from a previous encryption query with $\overline{[\mathbf{c}]} \neq \overline{[\mathbf{c}']}$, but $\mathsf{H}(\overline{[\mathbf{c}]}) = \mathsf{H}(\overline{[\mathbf{c}']})$. In that case we can straightforwardly employ $\mathcal{A}$ to obtain an adversary $\mathcal{B}$ attacking the collision resistance of H in time $T(\mathcal{B}) \approx T(\mathcal{A}) + (Q_{\mathbf{enc}} + Q_{\mathbf{dec}}) \cdot \mathsf{poly}(\lambda)$ for a polynomial poly independent of $T(\mathcal{A})$. Thereby we obtain an upper bound on the described event of $\text{Adv}^{\text{cr}}_{\mathsf{H},\mathcal{B}}(\lambda)$.

In the second event, $\mathcal{A}$ provides a valid decryption query $([\mathbf{c}], \Pi)$, such that $\overline{[\mathbf{c}]} = \overline{[\mathbf{c}']}$ for a previous challenge ciphertext $[\mathbf{c}'] \neq [\mathbf{c}]$. By the properties of **PS**, the proof corresponding to a ciphertext $[\mathbf{c}]$ is unique, which in particular implies $[\mathbf{c}] \notin \text{span}([\mathbf{A}])$. We bound the probability that $\mathcal{A}$ submits a valid decryption query $([\mathbf{c}], \Pi)$ such that $[\mathbf{c}] \notin \text{span}([\mathbf{A}])$ by $Q_{\mathbf{dec}} \cdot \text{uncert}_{\mathcal{A}}(\lambda)$, using a series of hybrids: For $i = 0, \ldots, Q_{\mathbf{dec}}$ let $\mathbf{G}_{0.i}$ be defined like $\mathbf{G}_0$, except $\mathcal{O}_{\mathbf{dec}}$ checks the freshness of τ for the first i queries and operates as in game $\mathbf{G}_0$ from the $(i+1)$-st query on. Note that game $\mathbf{G}_{0.0}$ equals $\mathbf{G}_0$ and game $\mathbf{G}_{0.Q_{\mathbf{dec}}}$ equals $\mathbf{G}_1$. We show that for all $i \in \{0, \ldots, Q_{\mathbf{dec}} - 1\}$:

$$|\varepsilon_{0.i} - \varepsilon_{0.(i+1)}| \leq \Pr_{K \leftarrow_R \mathcal{K}}[\mathsf{pred}_{i+1}(K) = 1].$$

Game $\mathbf{G}_{0.i}$ and game $\mathbf{G}_{0.(i+1)}$ only differ when the $(i+1)$-st query to $\mathcal{O}_{\mathbf{dec}}$ is valid with $\overline{[\mathbf{c}]} = \overline{[\mathbf{c}']}$ for a previous challenge ciphertext $[\mathbf{c}'] \neq [\mathbf{c}]$. As all challenge ciphertexts are in span$([\mathbf{A}])$, they do not reveal anything about $\mathbf{k}_0$ beyond the public key $[\mathbf{k}_0^\top \mathbf{A}]$. Thus, for $[\mathbf{c}] \notin \text{span}([\mathbf{A}])$, the value $\mathbf{k}_0^\top [\mathbf{c}]$ looks uniformly random from the adversary's point of view, proving the claimed distance between game $\mathbf{G}_{0.i}$ and game $\mathbf{G}_{0.(i+1)}$. Altogether we obtain

$$|\varepsilon_0 - \varepsilon_1| \leq \text{Adv}^{\text{cr}}_{\mathsf{H},\mathcal{B}}(\lambda) + Q_{\mathbf{dec}} \cdot \text{uncert}_{\mathcal{A}}(\lambda).$$

$\mathbf{G_1} \rightsquigarrow \mathbf{G_2}$**:** From $\mathbf{G}_2$ on, the way challenge ciphertexts are computed is changed. Namely, the simulation algorithmen $\mathbf{PSim}(psk, [\mathbf{c}])$ is used instead of $\mathbf{PPrv}(ppk, [\mathbf{c}], \mathbf{r})$ to compute $(\Pi, [\kappa])$. Since for all challenge ciphertexts we have $[\mathbf{c}] \in \mathcal{L}$, the proofs and keys are equal by the perfect zero-knowledge property of **PS**, and thus we have

$$\varepsilon_1 = \varepsilon_2.$$

$\mathbf{G_2} \rightsquigarrow \mathbf{G_3}$**:** Game $\mathbf{G}_3$ is like $\mathbf{G}_2$ except the vectors $[\mathbf{c}]$ in the challenge ciphertexts are chosen randomly in the span of $[\mathbf{A}_0]$.

We first employ the $Q_{\mathbf{enc}}$-fold $\mathcal{D}_{2k,k}$-MDDH assumption to tightly switch the vectors in the challenge ciphertexts from span($[\mathbf{A}]$) to uniformly random vectors over $\mathbb{G}^{2k}$. Next we use the $Q_{\mathbf{enc}}$-fold $\mathcal{U}_{2k,k}$-MDDH assumption to switch these vectors from random to $[\mathbf{A}_0\mathbf{r}]$.

To be specific, we build adversaries $\mathcal{B}$, $\mathcal{B}'$ such that for a polynomial poly independent of $T(\mathcal{A})$ we have $T(\mathcal{B}) \approx T(\mathcal{B}') \approx T(\mathcal{A}) + (Q_{\mathbf{enc}} + Q_{\mathbf{dec}}) \cdot \mathsf{poly}(\lambda)$ and

$$|\varepsilon_2 - \varepsilon_3| \leq \mathrm{Adv}^{Q_{\mathbf{enc}}\text{-mddh}}_{\mathbb{G},\mathcal{D}_{2k,k},\mathcal{B}}(\lambda) + \mathrm{Adv}^{Q_{\mathbf{enc}}\text{-mddh}}_{\mathbb{G},\mathcal{U}_{2k,k},\mathcal{B}'}(\lambda).$$

Let $([\mathbf{A}], [\mathbf{v}_1|\dots|\mathbf{v}_{Q_{\mathbf{enc}}}])$ with $[\mathbf{A}] \in \mathbb{G}^{2k\times k}$ and $[\mathbf{V}] := [\mathbf{v}_1|\dots|\mathbf{v}_{Q_{\mathbf{enc}}}] \in \mathbb{G}^{2k\times Q_{\mathbf{enc}}}$ be the $Q_{\mathbf{enc}}$-fold $\mathcal{D}_{2k,k}$-MDDH challenge received by $\mathcal{B}$. Then $\mathcal{B}$ samples $(ppk, psk) \leftarrow_R \mathbf{PGen}(1^\lambda)$, $\mathbf{k}_0, \mathbf{k}_1 \leftarrow_R \mathbb{Z}_p^{2k}$, $b \leftarrow_R \{0,1\}$ and sends the public key $pk := (ppk, [\mathbf{k}_0^\top\mathbf{A}], [\mathbf{k}_1^\top\mathbf{A}])$ to $\mathcal{A}$.

On the i-th query to $\mathcal{O}_{\mathbf{enc}}$, $\mathcal{B}$ sets the challenge ciphertext to $[\mathbf{c}] := [\mathbf{v}_i]$, next computes $\tau := \mathsf{H}([\mathbf{c}])$, $(\Pi, [\kappa]) := \mathbf{PSim}(psk, [\mathbf{v}_i])$ and finally $K_1 := (\mathbf{k}_0^\top + \tau\mathbf{k}_1^\top)[\mathbf{c}]$ (and $K_0 \leftarrow_R \mathcal{K}(\lambda)$ as usual). As $\mathcal{B}$ has generated the secret key itself, for decryption queries it can simply follow $\mathbf{KDec}(pk, sk, C)$.

In case $[\mathbf{V}] = [\mathbf{AR}]$, $\mathcal{B}$ perfectly simulates game $\mathbf{G}_2$. In case $[\mathbf{V}]$ is uniformly random over $\mathbb{G}^{2k\times Q_{\mathbf{enc}}}$, $\mathcal{B}$ simulates an intermediary game $\mathbf{H}$, where the challenge ciphertexts are chosen uniformly at random. Analogously we construct an adversary $\mathcal{B}'$ on the $Q_{\mathbf{enc}}$-fold $\mathcal{U}_{2k,k}$-MDDH assumption, who simulates game $\mathbf{H}$ if $[\mathbf{V}]$ is uniformly at random over $\mathbb{G}^{2k\times Q_{\mathbf{enc}}}$, and game $\mathbf{G}_3$, if $[\mathbf{V}] = [\mathbf{A}_0\mathbf{R}]$. Altogether this proves the claim stated above.

Finally, from Lemma 4 (random self-reducibility of $\mathcal{U}_{2k,k}$-MDDH), Lemma 3 ($\mathcal{D}_{2k,k}$-MDDH $\Rightarrow$ $\mathcal{U}_{2k,k}$-MDDH), and Lemma 2 (random self-reducibility of $\mathcal{D}_{2k,k}$-MDDH), we obtain an adversary $\mathcal{B}''$ such that $T(\mathcal{B}'') \approx T(\mathcal{A}) + (Q_{\mathbf{enc}} + Q_{\mathbf{dec}}) \cdot \mathsf{poly}(\lambda)$ where poly is independent of $T(\mathcal{A})$ and

$$|\varepsilon_2 - \varepsilon_3| \leq (1+k) \cdot \mathrm{Adv}^{\mathrm{mddh}}_{\mathbb{G},\mathcal{D}_{2k,k},\mathcal{B}''}(\lambda) + \frac{2}{p-1}.$$

$\mathbf{G_3} \rightsquigarrow \mathbf{G_4}$**:** We now restrict the adversary to decryption queries with $[\mathbf{c}] \in \mathrm{span}([\mathbf{A}])$. For the justification we refer to Lemma 9 .

$\mathbf{G_4} \rightsquigarrow \mathbf{G_5}$**:** In game $\mathbf{G}_5$, we change the keys $[\kappa]$ computed by $\mathcal{O}_{\mathbf{enc}}$ to random over $\mathbb{G}$. This is justified as follows.

Firstly, we can replace $\mathbf{k}_0$ by $\mathbf{k}_0 + \mathbf{A}^\perp\mathbf{u}$ with $\mathbf{u} \leftarrow_R \mathbb{Z}_p^k$ and $\mathbf{A}^\perp \in \mathsf{orth}(\mathbf{A})$, as those are identically distributed. Note that this change does neither affect the public key, nor the decryption queries, since for all $\mathbf{c} \in \mathrm{span}(\mathbf{A})$, $\mathbf{c}^\top(\mathbf{k}_0 + \mathbf{A}^\perp\mathbf{u}) = \mathbf{c}^\top\mathbf{k}_0$. Thus, the term $\mathbf{A}^\perp\mathbf{u}$ only shows up when $\mathcal{O}_{\mathbf{enc}}$ computes the value $[(\mathbf{A}^\perp\mathbf{u})^\top\mathbf{A}_0\mathbf{r}]$ for $\mathbf{r} \leftarrow_R \mathbb{Z}_p^k$ as part of the key K_1 (the key that is not chosen at random by the security experiment).

Secondly, the distributions $(\mathbf{A}^\perp\mathbf{u})^\top\mathbf{A}_0$ and $\mathbf{v}^\top \leftarrow_R \mathbb{Z}_p^{1\times k}$ are $1 - 2^{-\Omega(\lambda)}$-close.

Altogether, we obtain that $\mathcal{O}_{\mathbf{enc}}$, on its j-th query for each $j \in [Q_{\mathbf{enc}}]$, can compute key K_1 for $\mathbf{r}_j \leftarrow_R \mathbb{Z}_p^k$, and $\mathbf{v} \leftarrow_R \mathbb{Z}_p^k$ as

$$K_1 := \left[(\mathbf{k}_0 + \tau\mathbf{k}_1)^\top \mathbf{A}_0\mathbf{r}_j\right] + [\mathbf{v}^\top \mathbf{r}_j] + [\kappa].$$

We then switch from $([\mathbf{r}_j], [\mathbf{v}^\top \mathbf{r}_j])$ to $([\mathbf{r}_j], [z_j])$, where z_j is a uniformly random value over $\mathbb{G}$, using the $Q_{\mathbf{enc}}$-fold $\mathcal{U}_k$-MDDH assumption as follows. On input $([\mathbf{B}], [\mathbf{h}_1|\dots|\mathbf{h}_{Q_{\mathbf{enc}}}])$ with $\mathbf{B} \leftarrow_R \mathcal{U}_k$ (that is $\mathbf{B} \in \mathbb{Z}_p^{(k+1)\times k}$) and $\mathbf{h}_1, \dots, \mathbf{h}_{Q_{\mathbf{enc}}} \in \mathbb{Z}_p^{k+1}$, $\mathcal{B}$ samples $(ppk, psk) \leftarrow_R \mathbf{PGen}(1^\lambda)$, $\mathbf{k}_0, \mathbf{k}_1 \leftarrow_R \mathbb{Z}_p^{2k}$, $b \leftarrow_R \{0,1\}$ and sends the public key $pk := (ppk, [\mathbf{k}_0^\top \mathbf{A}], [\mathbf{k}_1^\top \mathbf{A}])$ to $\mathcal{A}$. In the following for all $j \in Q_{\mathbf{enc}}$ let $\overline{[\mathbf{h}_j]} \in \mathbb{G}^k$ comprise the upper k entries and $\underline{[\mathbf{h}_j]} \in \mathbb{G}$ the $(k+1)$-st entry of $[\mathbf{h}_j]$ and similar for $[\mathbf{B}]$ let $\overline{[\mathbf{B}]} \in \mathbb{G}^{k\times k}$ be the upper square matrix of $[\mathbf{B}]$ and $\underline{[\mathbf{B}]} \in \mathbb{G}^{1\times k}$ comprise the last row.

On the j-th encryption query, $\mathcal{B}$ sets $[\mathbf{c}] := \mathbf{A}_0\overline{[\mathbf{h}_j]}$ (and thus $[\mathbf{r}_j] := \overline{[\mathbf{h}_j]}$) and computes the key as

$$K_1 := \left[(\mathbf{k}_0 + \tau\mathbf{k}_1)^\top \mathbf{c}\right] + \underline{[\mathbf{h}_j]} + [\kappa].$$

The adversary $\mathcal{B}$ can answer decryption queries as usual using $\mathbf{k}_0$, as decryption queries outside $\mathcal{L}$ are rejected.

Now if $([\mathbf{B}], [\mathbf{h}_1|\dots|\mathbf{h}_{Q_{\mathbf{enc}}}])$ was a real $\mathcal{U}_k$-MDDH challenge, we have $\mathbf{h}_j = \mathbf{B}\mathbf{s}_j$ for a $\mathbf{s}_j \leftarrow_R \mathbb{Z}_p^k$ and thus we have $\mathbf{r}_j = \overline{\mathbf{B}}\mathbf{s}_j$ and $\underline{[\mathbf{h}_j]} = \underline{[\mathbf{B}]}\mathbf{s}_j = \underline{[\mathbf{B}]}\overline{\mathbf{B}}^{-1}\mathbf{r}_j$. Note that the distribution of $\underline{[\mathbf{B}]}\overline{\mathbf{B}}^{-1}$ is statistically close to the distribution of $\mathbf{v}^\top$ and therefore $\mathcal{B}$ simulates game $\mathbf{G}_4$. In case $\mathbf{h}_j$ was chosen uniformly at random from $\mathbb{Z}_p^{k+1}$, the adversary $\mathcal{B}$ simulates game $\mathbf{G}_5$ instead. In the end adversary $\mathcal{B}$ can thus forward the output of $\mathcal{A}$ to its own experiment.

Finally, Lemma 3, Lemma 4 and Lemma 5 yield the existence of an adversary $\mathcal{B}'$ such that $T(\mathcal{B}') \approx T(\mathcal{A}) + (Q_{\mathbf{enc}} + Q_{\mathbf{dec}}) \cdot \mathsf{poly}(\lambda)$ where poly is a polynomial independent of $T(\mathcal{A})$, and

$$|\varepsilon_4 - \varepsilon_5| \leq \mathrm{Adv}^{\mathrm{mddh}}_{\mathbb{G},\mathcal{D}_{2k,k},\mathcal{B}'}(\lambda) + 2^{-\Omega(\lambda)}.$$

Game $\mathbf{G_5}$: In this game, the keys K_1 computed by $\mathcal{O}_{\mathbf{enc}}$ are uniformly random, since the value $[\kappa]$ which shows up in $K_1 := [(\mathbf{k}_0 + \tau\mathbf{k}_1)^\top \mathbf{c}] + [\kappa]$ is uniformly random for each call to $\mathcal{O}_{\mathbf{enc}}$. The same holds true for the keys K_0 which are chosen at random from $\mathcal{K}(\lambda)$ throughout all games. Therefore, the output of $\mathcal{O}_{\mathbf{enc}}$ is now independent of the bit b chosen in $\mathrm{Exp}^{\mathrm{ccca}}_{\mathbf{KEM},\mathcal{A}}(\lambda)$. This yields

$$\varepsilon_5 = 0.$$

$\square$

Lemma 9. *The security games G_3 and G_4 defined for the proof of Theorem 3 (security of the KEM, see Figure 7) are computationally indistinguishable. More precisely, for every IND-CCCA adversary $\mathcal{A}$ that makes at most $Q_{\mathbf{enc}}$ encryption and $Q_{\mathbf{dec}}$ decryption queries, there exist adversaries $\mathcal{B}^{\mathrm{csnd}}$, $\mathcal{B}^{\mathrm{ind}}$, $\mathcal{B}^{\mathrm{mddh}}$ and $\mathcal{B}^{\widetilde{\mathrm{csnd}}}$ with running time $T(\mathcal{B}^{\mathrm{csnd}}) \approx T(\mathcal{B}^{\mathrm{ind}}) \approx T(\mathcal{B}^{\mathrm{mddh}}) \approx T(\mathcal{A}) + (Q_{\mathbf{enc}} + Q_{\mathbf{dec}}) \cdot \mathsf{poly}(\lambda)$ respectively $T(\mathcal{B}^{\widetilde{\mathrm{csnd}}}) \approx T(\mathcal{A}) + (Q_{\mathbf{enc}} + Q_{\mathbf{enc}} \cdot Q_{\mathbf{dec}}) \cdot \mathsf{poly}(\lambda)$, where poly is a polynomial independent of $T(\mathcal{A})$, and such that*

$$|\varepsilon_3 - \varepsilon_4| \leq \frac{1}{2} \cdot \mathrm{Adv}^{\mathrm{csnd}}_{\mathcal{L}_{\mathrm{snd}},\mathbf{PS},\mathcal{B}^{\mathrm{csnd}}}(\lambda) + \frac{1}{2} \cdot \mathrm{Adv}^{\mathrm{ind}}_{\mathcal{L}_{\mathrm{snd}},\mathbf{PS},\widetilde{\mathbf{PS}},\mathcal{B}^{\mathrm{ind}}}(\lambda)$$

$$+ 2\lambda \cdot \mathrm{Adv}^{\mathrm{mddh}}_{\mathbb{G},\mathcal{D}_{2k,k},\mathcal{B}^{\mathrm{mddh}}}(\lambda) + \frac{\lambda}{2} \cdot \mathrm{Adv}^{\mathrm{csnd}}_{\widetilde{\mathcal{L}_{\mathrm{snd}}},\widetilde{\mathbf{PS}},\mathcal{B}^{\widetilde{\mathrm{csnd}}}}(\lambda)$$
$$+ \frac{\lambda+2}{2} \cdot Q_{\mathbf{enc}} \cdot Q_{\mathbf{dec}} \cdot \mathrm{uncert}_{\mathcal{A}}(\lambda) + Q_{\mathbf{enc}} \cdot 2^{-\Omega(\lambda)}.$$

Proof. From game $\mathbf{G}_4$ on, decryption queries outside the span of $[\mathbf{A}]$ will always be answered with $\perp$ independently of the corresponding proof Π.

Games $\mathbf{G}_3$ and $\mathbf{G}_4$ behave the same, as long as an adversary $\mathcal{A}$ does not manage to submit a decryption query $(\mathsf{pred}, ([\mathbf{c}], \Pi))$ with $[\mathbf{c}] \notin \mathrm{span}([\mathbf{A}])$, on which $\mathcal{O}_{\mathbf{dec}}$ does not abort in $\mathbf{G}_3$.

In the following we will introduce probabilities conditioned on the bit b, which determines whether the encryption oracle returns uniformly random keys or real keys. Namely for $i \in \{3,4\}$ and $\beta \in \{0,1\}$ let $\varepsilon_{i|\beta}$ denote the probability that $\mathcal{A}$ wins game $\mathbf{G}_i$ under the condition that $b = \beta$ was drawn by the challenger. We prove that $\mathbf{G}_3$ and $\mathbf{G}_4$ are computationally indistinguishable, by a case analysis, depending on the bit b.

<u>For $b = 0$</u>: the encryption oracle $\mathcal{O}_{\mathbf{enc}}$ of the experiment $\mathrm{Exp}^{\mathrm{ind-ccca}}_{\mathbf{KEM},\mathcal{A}}(\lambda)$ returns keys chosen uniformly at random from $\mathcal{K}(\lambda)$, thus, all the adversary can information theoretically learn about $\mathbf{k}_0$ is $[\mathbf{k}_0^\top \mathbf{A}]$ from the public key. We can use the remaining entropy from $\mathbf{k}_0$ to argue that the adversary $\mathcal{A}$ can only submits queries $(\mathsf{pred}, ([\mathbf{c}], \Pi))$ to $\mathcal{O}_{\mathbf{dec}}$, for which the correpodsing key does not satisfies pred.

Namely, we replace $\mathbf{k}_0$ by $\mathbf{k}_0 + \mathbf{A}^\perp \mathbf{u}$ for $\mathbf{A}^\perp \in \mathsf{orth}(\mathbf{A})$, and $\mathbf{u} \leftarrow_R \mathbb{Z}_p^k$ as both are distributed identically. This change does not affect the public key, but for all $[\mathbf{c}] \notin \mathrm{span}([\mathbf{A}])$ we have: $[\mathbf{c}]^\top \mathbf{A}^\perp \neq \mathbf{0}$, and $[\mathbf{c}]^\top \mathbf{A}^\perp \mathbf{u}$ is uniformly random over $\mathbb{G}$. Therefore, the probability that the decryption oracle accepts a query $(\mathsf{pred}, ([\mathbf{c}], \Pi))$ with $[\mathbf{c}] \notin \mathrm{span}([\mathbf{A}])$, in $\mathbf{G}_3$ for $b = 0$, is bounded by $\Pr_{K \in \mathcal{K}}[\mathsf{pred}(K) = 1]$. Via a hybrid argument across all decryption queries, we obtain

$$|\varepsilon_{3|0} - \varepsilon_{4|0}| \leq Q_{\mathbf{dec}} \cdot \mathrm{uncert}_{\mathcal{A}}(\lambda).$$

<u>For $b = 1$</u>: In the following we will call a query *critical*, if it is of the form $(\mathsf{pred}, ([\mathbf{c}], \Pi))$ with $[\mathbf{c}] \notin \mathrm{span}([\mathbf{A}])$ and the decryption oracle does not abort in the respective game. Our goal is to bound the event of $\mathcal{A}$ submitting such a query. More precisely, we give the corresponding game $\mathbf{H}_0$ in Fig. 8, where $\mathcal{A}$ gets the public key pk as input and access to the oracles $\mathcal{O}_{\mathbf{enc}}$ and $\mathcal{O}_{\mathbf{dec}}$. $\mathcal{A}$ wins if the decryption oracle returns *critical query* at some point. Note that except for the altered winning condition, the oracles behave as in game $\mathbf{G}_3$ for $b = 1$. We denote the probability that the adversary $\mathcal{A}$ wins game $\mathbf{H}_{\mathbf{x}}$ by $\varepsilon_{\mathbf{H}.\mathbf{x}}$. Note that we have

$$|\varepsilon_{3|1} - \varepsilon_{4|1}| \leq \varepsilon_{\mathbf{H}.0}$$

and thus altogether we obtain

$$|\varepsilon_3 - \varepsilon_4| \leq \frac{1}{2} \cdot \left(\varepsilon_{\mathbf{H}.0} + Q_{\mathbf{dec}} \cdot \mathrm{uncert}_{\mathcal{A}}(\lambda)\right).$$

In the following we will bound $\varepsilon_{\mathbf{H}.0}$ via a sequence of games. We give an overview of the games in Fig. 9.

We will always assume that the freshness of τ is checked by the decryption oracle (and the query is answered with $\perp$ if it fails). In all games, an adversary wins if it manages to submit a critical query.

$\underline{\mathrm{Exp}^{\mathbf{H}_x}_{\mathbf{KEM},\mathcal{A}}(\lambda)}$:
$(pk, sk) \leftarrow_R \mathbf{KGen}(1^\lambda)$
$\mathbf{v} \leftarrow_R \mathbb{Z}_p^{2k}$
$\mathcal{C}_{\mathbf{enc}} := \emptyset$
$\mathcal{A}^{\mathcal{O}_{\mathbf{enc}},\mathcal{O}_{\mathbf{dec}}(\cdot,\cdot)}(pk)$
if $\mathcal{O}_{\mathbf{dec}}$ **returned** *critical query*
  return 1
else return 0

$\underline{\mathcal{O}_{\mathbf{enc}}}$:
$\mathbf{r} \leftarrow_R \mathbb{Z}_p^k$
$[\mathbf{c}] := [\mathbf{A}_0]\mathbf{r}$
$\tau := \mathsf{H}(\overline{[\mathbf{c}]})$
$(\Pi, [\kappa]) := \mathbf{PSim}(ppk, psk, [\mathbf{c}])$
$C := ([\mathbf{c}], \Pi)$
$K := \left(\mathbf{k}_0 + \tau\mathbf{k}_1 + \mathbf{v}\right)^\top [\mathbf{c}] + [\kappa]$
$\mathcal{C}_{\mathbf{enc}} := \mathcal{C}_{\mathbf{enc}} \cup \{C\}$
return (C, K)

$\underline{\mathcal{O}_{\mathbf{dec}}(\mathsf{pred}, ([\mathbf{c}], \Pi))}$:
$(v, [\kappa]) := \mathbf{PVer}(psk, [\mathbf{c}], \Pi)$
$\tau := \mathsf{H}(\overline{[\mathbf{c}]})$
if $([\mathbf{c}], \Pi) \notin \mathcal{C}_{\mathbf{enc}}$ **and** $v = 1$ **and** τ is fresh
  if $[\mathbf{c}] \in \mathrm{span}([\mathbf{A}])$
    $K := (\mathbf{k}_0 + \tau\mathbf{k}_1)^\top [\mathbf{c}] + [\kappa]$
    if $\mathsf{pred}(K) = 1$
      return K
  else $\boxed{\textbf{if } [\mathbf{c}] \in \mathrm{span}([\mathbf{A}_0])}$
    $K := \left(\mathbf{k}_0 + \tau\mathbf{k}_1 + \mathbf{v}\right)^\top [\mathbf{c}] + [\kappa]$
    if $\mathsf{pred}(K) = 1$
      return *critical query* **and abort**
return $\perp$

Fig. 8: Games $\mathbf{H}_0$, $\boxed{\mathbf{H}_1}$ and $\mathbf{H}_2$

$\mathbf{H_0} \rightsquigarrow \mathbf{H_1}$**:** We will first reject decryption queries outside $\mathcal{L}^{\mathrm{snd}}$. We justify this employing the constrained soundness of **PS**. Let $\mathcal{A}$ be an adversary distinguishing between games $\mathbf{H}_0$ and $\mathbf{H}_1$, that is an adversary submitting a succesful decryption query outside $\mathcal{L}_{\mathrm{snd}}$ in $\mathbf{H}_0$. Then we construct an adversary $\mathcal{B}$ breaking constrained $\mathcal{L}_{\mathrm{snd}}$-soundness of **PS** as follows.

On receiving the public key ppk of **PS**, the adversary $\mathcal{B}$ samples $\mathbf{k}_0, \mathbf{k}_1 \leftarrow_R \mathbb{Z}_p^{2k}$, and sends $pk := (ppk, [\mathbf{k}_0^\top \mathbf{A}], [\mathbf{k}_1^\top \mathbf{A}])$ to $\mathcal{A}$.

On an encryption query of $\mathcal{A}$, the adversary $\mathcal{B}$ can employ its simulation oracle $\mathcal{O}_{\mathbf{sim}}$ to obtain $([\mathbf{c}], \Pi, [\kappa])$ with $[\mathbf{c}] \in \mathrm{span}([\mathbf{A}_0])$. The adversary $\mathcal{B}$ now computes $\tau := \mathsf{H}(\overline{[\mathbf{c}]})$ and sets $C := ([\mathbf{c}], \Pi)$ and $K := (\mathbf{k}_0 + \tau\mathbf{k}_1)^\top [\mathbf{c}] + [\kappa]$. Finally $\mathcal{B}$ returns (C, K) to $\mathcal{A}$.

To answer $\mathcal{A}$'s queries to $\mathcal{O}_{\mathbf{dec}}$ of the form $(\mathsf{pred}, ([\mathbf{c}], \Pi))$, we distinguish the following cases, where we use that $\mathcal{B}$ has access to $\mathbf{A}$ and $\mathbf{A}_0$. In all cases $\mathcal{B}$ computes $\tau := \mathbf{H}(\overline{[\mathbf{c}]})$ and defines the predicate $\mathsf{pred}' \colon K \mapsto \mathsf{pred}((\mathbf{k}_0 + \tau\mathbf{k}_1)^\top[\mathbf{c}] + K)$. Next $\mathcal{B}$ queries $\mathcal{O}_{\mathbf{ver}}$ on $([\mathbf{c}], \Pi, \mathsf{pred}')$.

In case $[\mathbf{c}] \in \operatorname{span}([\mathbf{A}])$, the oracle returns either $\perp$ or a key $[\kappa]$ to $\mathcal{B}$. In the former case $\mathcal{B}$ forwards $\perp$ to $\mathcal{A}$, in the latter the key $K := (\mathbf{k}_0 + \tau\mathbf{k}_1)^\top[\mathbf{c}] + [\kappa]$.

If $[\mathbf{c}] \in \operatorname{span}([\mathbf{A}_0])$, the oracle $\mathcal{O}_{\mathbf{ver}}$ returns either $\perp$ or the adversary $\mathcal{B}$ has lost the constrained soundness game. In the former case, $\mathcal{B}$ forwards $\perp$ to $\mathcal{A}$. In the latter case the adversary $\mathcal{A}$ managed to submit a critical query in both games $\mathbf{H}_0$ and $\mathbf{H}_1$ and thus did not succeed in distinguishing between the two.

Finally, if $[\mathbf{c}] \notin \operatorname{span}([\mathbf{A}]) \cup \operatorname{span}([\mathbf{A}_0])$, the oracle $\mathcal{O}_{\mathbf{ver}}$ returns either $\perp$ (in which case $\mathcal{B}$ sends $\perp$ to $\mathcal{A}$), or the adversary $\mathcal{B}$ has win the constrained soundness game. Only in the last case does $\mathcal{A}$ distinguish between $\mathbf{H}_0$ and $\mathbf{H}_1$.

Altogether we obtain an adversary $\mathcal{B}$ breaking the constrained $\mathcal{L}_{\mathrm{snd}}$-soundness of $\mathbf{PS}$ in time $T(\mathcal{B}) \approx T(\mathcal{A}) + (Q_{\mathbf{enc}} + Q_{\mathbf{dec}}) \cdot \mathsf{poly}(\lambda)$, where poly is a polynomial independent of $T(\mathcal{A})$, such that

$$|\varepsilon_{\mathbf{H}.0} - \varepsilon_{\mathbf{H}.1}| \leq \mathrm{Adv}^{\mathrm{csnd}}_{\mathcal{L}_{\mathrm{snd}},\mathbf{PS},\mathcal{B}}(\lambda).$$

$\mathbf{H_1} \rightsquigarrow \mathbf{H_2}$**:** We alter the oracles in game $\mathbf{H}_2$ as described in Fig. 8, where the same $\mathbf{v} \leftarrow_R \mathbb{Z}_p^{2k}$ is used across all oracle calls. The appearance of the extra random term $\mathbf{v}$ in encryption and decryption queries with $[\mathbf{c}] \in \operatorname{span}([\mathbf{A}_0])$ is justified as follows.

In an intermediary game we first replace $\mathbf{k}_0$ by $\mathbf{k}_0 + \mathbf{A}^\perp\mathbf{u}$, where $\mathbf{A}^\perp \in \mathsf{orth}(\mathbf{A})$ and $\mathbf{u} \leftarrow_R \mathbb{Z}_p^k$. This transition does not change the view of the adversaries as the keys $\mathbf{k}_0$ and $\mathbf{k}_0 + \mathbf{A}^\perp\mathbf{u}$ are both distributed uniformly random over $\mathbb{Z}_p^{2k}$. Note that this change neither affects the public key, nor the keys computed by $\mathcal{O}_{\mathbf{dec}}$ when queried on inputs containing $[\mathbf{c}] \in \operatorname{span}([\mathbf{A}])$, since $(\mathbf{k}_0 + \mathbf{A}^\perp\mathbf{u})^\top[\mathbf{c}] = \mathbf{k}_0^\top[\mathbf{c}]$.

Next for $\mathbf{A}_0^\perp \in \mathsf{orth}(\mathbf{A}_0)$ and $\mathbf{u}_0 \leftarrow_R \mathbb{Z}_p^k$ we replace $\mathbf{k}_0 + \mathbf{A}^\perp\mathbf{u}$ by $\mathbf{k}_0 + \mathbf{A}^\perp\mathbf{u} + \mathbf{A}_0^\perp\mathbf{u}_0$ in all encryption queries and decryption queries with $[\mathbf{c}] \in \operatorname{span}([\mathbf{A}_0])$, which does not change the adversary's view, since we have $(\mathbf{A}^\perp\mathbf{u})^\top[\mathbf{c}] = (\mathbf{A}^\perp\mathbf{u} + \mathbf{A}_0^\perp\mathbf{u}_0)^\top[\mathbf{c}]$.

With probability $1 - 2^{-\Omega(\lambda)}$ over the choices of $\mathbf{A}, \mathbf{A}_0$ the column vectors of $\mathbf{A}^\perp$ and $\mathbf{A}_0^\perp$ together form a basis of $\mathbb{Z}_p^{2k}$, and thus $\mathbf{A}^\perp\mathbf{u} + \mathbf{A}_0^\perp\mathbf{u}_0$ is distributed uniformly random over $\mathbb{Z}_p^{2k}$ with overwhelming probability and can be replaced by $\mathbf{v} \leftarrow \mathbb{Z}_p^{2k}$.

This yields

$$|\varepsilon_{\mathbf{H}.1} - \varepsilon_{\mathbf{H}.2}| \leq 2^{-\Omega(\lambda)}.$$

$\mathbf{H_2} \rightsquigarrow \mathbf{H_3}$**:** By the $\widetilde{\mathcal{L}_{\mathrm{snd}}}$-extensibility of $\mathbf{PS}$, there exists a proof system $\widetilde{\mathbf{PS}}$, such that $\mathbf{PS}$ and $\widetilde{\mathbf{PS}}$ are $\mathcal{L}_{\mathrm{snd}}$-indistinguishable. From game $\mathbf{H}_3$ on, we replace $\mathbf{PS}$ by $\widetilde{\mathbf{PS}}$.

From an adversary $\mathcal{A}$ distinguishing between those to games, we can construct an adversary $\mathcal{B}$ breaking the $\mathcal{L}_{\mathrm{snd}}$-indistinguishability as follows, where $\mathcal{B}$ has either access to the oracles $\mathcal{O}^0_{\mathbf{sim}}$ and $\mathcal{O}^0_{\mathbf{ver}}$ of $\mathbf{PS}$, or to the oracles $\mathcal{O}^1_{\mathbf{sim}}$ and $\mathcal{O}^1_{\mathbf{ver}}$ of $\widetilde{\mathbf{PS}}$ and has to distinguish between the two cases.

Note that we do not change the distribution of $[\mathbf{c}]$ in simulation queries in this step, that is in both games $[\mathbf{c}]$ is chosen uniformly at random from $\operatorname{span}([\mathbf{A}_0])$.

#	proof system	ch. $\mathbf{k}_{\Delta}^{\text{enc}}(\tau)$	$\mathbf{k}_{\Delta}^{\text{dec}}(\tau, [\mathbf{c}])$ used by $\mathcal{O}_{\text{dec}}$ on $[\mathbf{c}]$ for which $[\mathbf{c}]^{\top}\mathbf{A}^{\perp} \neq [0]$	$\mathcal{O}_{\text{dec}}$ checks	game knows	remark
$\mathbf{H}_0$	**PS**	0	0		$\mathbf{A}$	
$\mathbf{H}_1$	**PS**	0	0	$[\mathbf{c}] \in \mathcal{L}_{\text{snd}}$	$\mathbf{A}$, $\mathbf{A}_0, \mathbf{A}_1$	$\mathcal{L}_{\text{snd}}$-soundness
$\mathbf{H}_2$	**PS**	$\mathbf{v}$	$\mathbf{v}$	$[\mathbf{c}] \in \mathcal{L}_{\text{snd}}$	$\mathbf{A}, \mathbf{A}_0, \mathbf{A}_1$	statistical
$\mathbf{H}_3$	$\widetilde{\mathbf{PS}}$	$\mathbf{v}$	$\mathbf{v}$	$[\mathbf{c}] \in \mathcal{L}_{\text{snd}}$	$\mathbf{A}, \mathbf{A}_0, \mathbf{A}_1$	$\widetilde{\mathcal{L}_{\text{snd}}}$-extensibility
$\mathbf{H}_4$	$\widetilde{\mathbf{PS}}$	$\mathbf{v}$	$\mathbf{v}$		$\mathbf{A}$	win. chances increase
$\mathbf{H}_5$	$\widetilde{\mathbf{PS}}$	$\mathbf{F}(\tau)$	$\{\mathbf{F}(\tau^{(j)})\}$		$\mathbf{A}$	see Figure 10

Fig. 9: Security of the KEM. Column "**proof system**" describes the underlying proof system used, where $\widetilde{\mathbf{PS}}$ is a $\widetilde{\mathcal{L}_{\text{snd}}}$-qualified proof system, such that **PS** and $\widetilde{\mathbf{PS}}$ are $\mathcal{L}_{\text{snd}}$-indistinguishable. Column "**ch.** $\mathbf{k}_{\Delta}^{\text{enc}}(\tau)$" refers to the vector $\mathbf{k}_{\Delta}^{\text{enc}}(\tau)$ used by $\mathcal{O}_{enc}$ when computing the key $K := [(\mathbf{k}_0 + \tau\mathbf{k}_1 + \mathbf{k}_{\Delta}^{\text{enc}}(\tau))^{\top}\mathbf{c}] + [\kappa]$ for challenge ciphertexts. $\mathbf{v}$ denotes a value in $\mathbb{Z}_p^{2k}$ chosen uniformly random, $\mathbf{F}\colon \{0,1\}^{\lambda} \to \mathbb{Z}_p^{2k}$ denotes a random function and $\tau := \mathsf{H}(\overline{[\mathbf{c}]})$. In the next column, we describe $\mathbf{k}_{\Delta}^{\text{dec}}(\tau, [\mathbf{c}])$ used by $\mathcal{O}_{dec}$ when computing the set of valid keys $\mathcal{S}_K := \left\{ \left(\mathbf{k}_0 + \tau\mathbf{k}_1 + \mathbf{k}_{\Delta}^{\text{dec}}(\tau^{(j)}, [\mathbf{c}])\right)^{\top} [\mathbf{c}] + [\kappa] \,\middle|\, \tau^{(j)} \in \mathcal{Q}_{\mathbf{dec}} \right\}$ on queries containing $[\mathbf{c}]$ such that $\mathbf{c}^{\top}\mathbf{A}^{\perp} \neq 0$. Here $\tau^{(j)} \in \mathcal{Q}_{\mathbf{dec}}$ for $j \in \{1, \ldots, Q_{\mathbf{enc}}\}$ denotes the tag from the j-th encryption query. By the set notation we want to imply that the decryption oracle accepts a predicate if it evaluates to 1 on any key in $\mathcal{S}_K$. The column "$\mathcal{O}_{\mathbf{dec}}$ **checks**" refers to additional checks performed on decryption queries ahead of decryption. We always assume $\mathcal{O}_{\mathbf{dec}}$ checks the freshness of τ and therefore not list it explicitely in the table. In case any of the checks fails, $\mathcal{O}_{\mathbf{dec}}$ returns $\perp$. The column "**game knows**" refers to what the game must know with respect to $\mathbf{A}$, $\mathbf{A}_0$ and $\mathbf{A}_1$.

#	**ch.** $[\mathbf{c}]$	**ch.** $\mathbf{k}_{\Delta}^{\text{enc}}(\tau)$	$\mathbf{k}_{\Delta}^{\text{dec}}(\tau,[\mathbf{c}])$ **used by** $\mathcal{O}_{\text{dec}}$ **on** $[\mathbf{c}]$ **for which** $[\mathbf{c}]^{\top}\mathbf{A}^{\perp} \neq [0]$	$\mathcal{O}_{\text{dec}}$ **checks**	**game knows**	**remark**
$\mathbf{H}_{4.i.0}$	$[\mathbf{A}_0]$	$\mathbf{F}_i(\tau_{\|i})$	$\left\{\mathbf{F}_i(\tau^{(j)}_{\|i})\right\}$		$\mathbf{A}$	$\mathbf{H}_{4.0.0} = \mathbf{H}_4$
$\mathbf{H}_{4.i.1}$	$[\mathbf{A}_{\tau_{i+1}}]$	$\mathbf{F}_i(\tau_{\|i})$	$\left\{\mathbf{F}_i(\tau^{(j)}_{\|i})\right\}$		$\mathbf{A}$	$\mathcal{D}_{2k,k}$-MDDH
$\mathbf{H}_{4.i.2}$	$[\mathbf{A}_{\tau_{i+1}}]$	$\mathbf{F}_i(\tau_{\|i})$	$\left\{\mathbf{F}_i(\tau^{(j)}_{\|i})\right\}$	$[\mathbf{c}] \in \widetilde{\mathcal{L}_{\text{snd}}}$	$\mathbf{A}$, $\mathbf{A}_0, \mathbf{A}_1$	$\widetilde{\mathcal{L}_{\text{snd}}}$-soundness
$\mathbf{H}_{4.i.3}$	$[\mathbf{A}_{\tau_{i+1}}]$	$\underline{\tau_{i+1} = 0:}$ $\mathbf{A}_0^{\perp}\widetilde{\mathbf{F}}_i^{(0)}(\tau_{\|i}) + \mathbf{A}_1^{\perp}\mathbf{F}_i^{(1)}(\tau_{\|i})$ $\underline{\tau_{i+1} = 1:}$ $\mathbf{A}_0^{\perp}\mathbf{F}_i^{(0)}(\tau_{\|i}) + \mathbf{A}_1^{\perp}\widetilde{\mathbf{F}}_i^{(1)}(\tau_{\|i})$	$\underline{\text{if } [\mathbf{c}] \in \text{span}([\mathbf{A}_0]):}$ $\left\{\mathbf{A}_0^{\perp}\widetilde{\mathbf{F}}_i^{(0)}(\tau^{(j)}_{\|i}) + \mathbf{A}_1^{\perp}\mathbf{F}_i^{(1)}(\tau^{(j)}_{\|i})\right\}$ $\underline{\text{if } [\mathbf{c}] \in \text{span}([\mathbf{A}_1]):}$ $\left\{\mathbf{A}_0^{\perp}\mathbf{F}_i^{(0)}(\tau^{(j)}_{\|i}) + \mathbf{A}_1^{\perp}\widetilde{\mathbf{F}}_i^{(1)}(\tau^{(j)}_{\|i})\right\}$	$[\mathbf{c}] \in \widetilde{\mathcal{L}_{\text{snd}}}$	$\mathbf{A}, \mathbf{A}_0, \mathbf{A}_1$	change of basis
$\mathbf{H}_{4.i.4}$	$[\mathbf{A}_{\tau_{i+1}}]$	$\mathbf{F}_{i+1}(\tau_{\|i+1})$	$\left\{\mathbf{F}_{i+1}(\tau^{(j)}_{\|i} d_{[\mathbf{c}]})\right\}$	$[\mathbf{c}] \in \widetilde{\mathcal{L}_{\text{snd}}}$	$\mathbf{A}, \mathbf{A}_0, \mathbf{A}_1$	conceptual
$\mathbf{H}_{4.i.5}$	$[\mathbf{A}_{\tau_{i+1}}]$	$\mathbf{F}_{i+1}(\tau_{\|i+1})$	$\left\{\mathbf{F}_{i+1}(\tau^{(j)}_{\|i} d_{[\mathbf{c}]})\right\}$		$\mathbf{A}, \mathbf{A}_0, \mathbf{A}_1$	win. chances increase
$\mathbf{H}_{4.i.6}$	$[\mathbf{A}_{\tau_{i+1}}]$	$\mathbf{F}_{i+1}(\tau_{\|i+1})$	$\left\{\mathbf{F}_{i+1}(\tau^{(j)}_{\|i} b), b \in \{0,1\}\right\}$		$\mathbf{A}$	win. chances increase
$\mathbf{H}_{4.i.7}$	$[\mathbf{A}_{\tau_{i+1}}]$	$\mathbf{F}_{i+1}(\tau_{\|i+1})$	$\left\{\mathbf{F}_{i+1}(\tau^{(j)}_{\|i+1})\right\}$		$\mathbf{A}$	$\mathbf{F}$ hard to guess on non-queried values

Fig. 10: Hybrid Games for Randomization. Columns are almost according to Figure 9. Additionally column "**ch.** $[\mathbf{c}]$" refers to the vector computed by $\mathcal{O}_{enc}$ as part of the challenge ciphertexts, where $\mathbf{A}$ indicates that $\mathbf{c} \leftarrow_R \text{span}(\mathbf{A})$, for instance. For $i = 0, \dots, \lambda$ by $\mathbf{F}_i : \{0,1\}^i \to \mathbb{Z}_p^{2k}$ and further by $\mathbf{F}_i^{(0)}, \mathbf{F}_i^{(1)}, \widetilde{\mathbf{F}}_i^{(0)}, \widetilde{\mathbf{F}}_i^{(1)} : \{0,1\}^i \to \mathbb{Z}_p^k$ we denote random functions, such that for all $\rho \in \{0,1\}^i$ and for a choice $\mathbf{A}_0^{\perp} \in \mathsf{orth}([\mathbf{A}_0])$ and $\mathbf{A}_1^{\perp} \in \mathsf{orth}([\mathbf{A}_1])$ we have $\mathbf{F}_i(\rho) = \mathbf{A}_0^{\perp}\mathbf{F}_i^{(0)}(\rho) + \mathbf{A}_1^{\perp}\mathbf{F}_i^{(1)}(\rho)$. Apart from this relation we require the functions to be independent. We set $d_{[\mathbf{c}]} = 0$ if $[\mathbf{c}] \in \text{span}([\mathbf{A}_0])$ and $d_{[\mathbf{c}]} = 1$ if $[\mathbf{c}] \in \text{span}([\mathbf{A}_1])$. We always assume $\mathcal{O}_{\mathbf{dec}}$ checks the freshness of τ and therefore do not list it explicitly in the table. In case any of the checks fails, $\mathcal{O}_{\mathbf{dec}}$ returns $\perp$.

On receiving the public key ppk of $\mathbf{PS}$, the adversary $\mathcal{B}$ samples $\mathbf{k}_0, \mathbf{k}_1 \leftarrow_R \mathbb{Z}_p^{2k}$, and sends $pk := (ppk, [\mathbf{k}_0^\top \mathbf{A}], [\mathbf{k}_1^\top \mathbf{A}])$ to $\mathcal{A}$. Now $\mathcal{B}$ can employ its simulation oracle $\mathcal{O}_{\mathbf{sim}}^{\beta}$ to answer decryption queries.

To answer $\mathcal{A}$'s queries to $\mathcal{O}_{\mathbf{dec}}$ of the form $(\mathsf{pred}, ([\mathbf{c}], \Pi))$, we distinguish the following cases, where we use that $\mathcal{B}$ has access to $\mathbf{A}$ and $\mathbf{A}_0$. All queries outside of $\mathcal{L}_{\mathrm{snd}}$ to the decryption oracle are answered with $\perp$ by $\mathcal{B}$. In case $[\mathbf{c}] \in \mathcal{L}_{\mathrm{snd}}$ the adversary $\mathcal{B}$ computes $\tau := \mathsf{H}(\overline{[\mathbf{c}]})$ and defines the predicate $\mathsf{pred}' \colon K \mapsto \mathsf{pred}((\mathbf{k}_0 + \tau \mathbf{k}_1)^\top [\mathbf{c}] + K)$. Next $\mathcal{B}$ queries $\mathcal{O}_{\mathbf{ver}}^{\beta}$ on $([\mathbf{c}], \Pi, \mathsf{pred}')$, to get either a key $[\kappa]$, or $\perp$. In the former case, $\mathcal{B}$ checks if $[\mathbf{c}] \in \mathrm{span}([\mathbf{A}])$, if this is the case, it returns the key $K := (\mathbf{k}_0 + \tau \mathbf{k}_1)^\top [\mathbf{c}] + [\kappa]$ to $\mathcal{A}$, if this is not the case it returns *critical query*, and ends the game. In the latter case, $\mathcal{B}$ sends $\perp$ to $\mathcal{A}$.

The adversary $\mathcal{B}$ now simulates game $\mathbf{H}_2$ in case $\beta = 0$ and game $\mathbf{H}_3$ in case $\beta = 1$, thus $\mathcal{B}$ can forward the output of $\mathcal{A}$ to its experiment.

Altogether we obtain thus an adversary $\mathcal{B}$ breaking the $\mathcal{L}_{\mathrm{snd}}$-indistinguishability of $\mathbf{PS}$ and $\widetilde{\mathbf{PS}}$ in time $T(\mathcal{B}) \approx T(\mathcal{A}) + (Q_{\mathbf{enc}} + Q_{\mathbf{dec}}) \cdot \mathsf{poly}(\lambda)$, where poly is a polynomial independent of $T(\mathcal{A})$, such that

$$|\varepsilon_{\mathbf{H}.2} - \varepsilon_{\mathbf{H}.3}| \leq \mathrm{Adv}^{\text{PS-ind}}_{\mathcal{L}_{\mathrm{snd}}, \mathbf{PS}, \widetilde{\mathbf{PS}}, \mathcal{B}}(\lambda),$$

$\mathbf{H_3} \rightsquigarrow \mathbf{H_4}$**:** From game $\mathbf{H}_4$ on, we again allow decryption queries outside $\mathcal{L}_{\mathrm{snd}}$. This can only increase the winning chances of the adversary, as it does not change the view on non-critical queries. We thus have

$$\varepsilon_{\mathbf{H}.3} \leq \varepsilon_{\mathbf{H}.4}.$$

$\mathbf{H_4} \rightsquigarrow \mathbf{H_5}$ To justify the transition from game $\mathbf{H}_4$ to game $\mathbf{H}_5$ we employ a hybrid argument comprising a number of games. We give an overview of these games in Fig. 10 and prove the reduction in the following.

$\mathbf{H}_{\mathbf{4}.\boldsymbol{i}.\mathbf{0}}$**:** For $i = 0, \ldots, \lambda$, in $\mathbf{H}_{4.i.0}$ the adversary has access to the oracles $\mathcal{O}_{\mathbf{enc}}$ and $\mathcal{O}_{\mathbf{dec}}$ defined as described in Fig. 11, where by $\mathbf{F}_i : \{0,1\}^i \rightarrow \mathbb{Z}_p^{2k}$ we denote a random function applied to the first i bits $\tau_{|i}$ of τ.

Note that in previous games ($\mathbf{H}_0$ to $\mathbf{H}_4$), for a statement $[\mathbf{c}] \notin \mathrm{span}([\mathbf{A}])$, $\mathcal{O}_{\mathbf{dec}}(\mathsf{pred}, ([\mathbf{c}], \Pi))$ computes one key K when the proof Π is valid, and return this key if $\mathsf{pred}(K) = 1$.

In game $\mathbf{H}_{4.i.0}$, instead, the decryption oracle will accept a query $(\mathsf{pred}, ([\mathbf{c}], \Pi))$ outside $\mathrm{span}([\mathbf{A}])$ as critical, if additionally to a valid proof Π, the corresponding predicate pred evaluates to 1 on any of the keys in the set

$$\mathcal{S}_K := \left\{ \left[\left(\mathbf{k}_0 + \tau^\star \mathbf{k}_1 + \mathbf{F}_i(\tau_{|i}) \right)^\top \mathbf{c} \right] + [\kappa] \,\middle|\, \tau \in \mathcal{Q}_{\mathbf{enc}} \right\},$$

where $\tau^\star := \mathsf{H}(\overline{[\mathbf{c}]})$ and $\mathcal{Q}_{\mathbf{enc}}$ denotes the set of tags previously computed by $\mathcal{O}_{\mathbf{enc}}$. As for $i = 0$ the function $\mathbf{F}_i = \mathbf{F}_0$ is a constant random value in $\mathbb{Z}_p^{2k}$, independent from its input τ, we have $\mathbf{H}_{4.0.0} = \mathbf{H}_4$. Also note that $\mathbf{H}_{4.\lambda.0} = \mathbf{H}_5$.

$\mathbf{H}_{\mathbf{4}.\boldsymbol{i}.\mathbf{0}} \rightsquigarrow \mathbf{H}_{\mathbf{4}.\boldsymbol{i}.\mathbf{1}}$**:** For $i = 0, \ldots, \lambda - 1$, $\mathbf{H}_{4.i.1}$ is defined as $\mathbf{H}_{4.i.0}$ except $\mathcal{O}_{\mathbf{enc}}$ computes ciphertexts of the form $[\mathbf{c}] := [\mathbf{A}_{\tau_{i+1}} \mathbf{r}]$, where τ_{i+1} denotes the $(i+1)$-st bit of τ, instead of $[\mathbf{A}_0 \mathbf{r}]$ in $\mathbf{H}_{4.i.0}$. We justify this transition by applying the $\mathcal{U}_{2k,k}$-MDDH assumption twice. First we use it once with

$\underline{\mathcal{O}_{\mathbf{enc}}}$:
$\mathbf{r} \leftarrow_R \mathbb{Z}_p^k$
$[\mathbf{c}] := [\mathbf{A}_0]\mathbf{r}$
$\tau := \mathsf{H}(\overline{[\mathbf{c}]})$
$(\Pi, [\kappa]) := \widetilde{\mathbf{PSim}}(ppk, psk, [\mathbf{c}])$
$C := ([\mathbf{c}], \Pi)$
$K := \left(\mathbf{k}_0 + \tau \mathbf{k}_1 + \boxed{\mathbf{F}_i(\tau_{|i})}\right)^\top [\mathbf{c}] + [\kappa]$
$\mathcal{C}_{\mathbf{enc}} := \mathcal{C}_{\mathbf{enc}} \cup \{C\}$
return (C, K)

$\underline{\mathcal{O}_{\mathbf{dec}}(\mathsf{pred}, ([\mathbf{c}], \Pi))}$:
$(v, [\kappa]) := \widetilde{\mathbf{PVer}}(psk, [\mathbf{c}], \Pi)$
$\tau^\star := \mathsf{H}(\overline{[\mathbf{c}]})$
if $([\mathbf{c}], \Pi) \notin \mathcal{C}_{\mathbf{enc}}$ **and** $v = 1$ **and** τ is fresh
    if $[\mathbf{c}] \in \mathrm{span}([\mathbf{A}])$
        $K := (\mathbf{k}_0 + \tau^\star \mathbf{k}_1)^\top [\mathbf{c}] + [\kappa]$
        if $\mathsf{pred}(K) = 1$
            return K
    else
        $\mathcal{S}_K := \left\{ \left(\mathbf{k}_0 + \tau^\star \mathbf{k}_1 + \boxed{\mathbf{F}_i(\tau_{|i})}\right)^\top [\mathbf{c}] + [\kappa] \,\middle|\, \tau \in \mathcal{Q}_{\mathbf{enc}} \right\}$
        if $\exists K \in \mathcal{S}_K$ **such that** $\mathsf{pred}(K) = 1$
            return *critical query* **and abort**
return $\perp$

Fig. 11: Oracles in Game $\mathbf{H}_{4.i.0}$

respect to $[\mathbf{A}_0]$ to tightly switch vectors from $[\mathbf{A}_0\mathbf{r}]$ to uniform random vectors over $\mathbb{G}^{2k}$. For the next step first note that a $\mathcal{U}_{2k,k}$-MDDH challenge $([\mathbf{A}_0], [\mathbf{v}])$ can be efficiently transformed into a $\mathcal{U}_{2k,k}$-MDDH challenge $([\mathbf{A}_1], [\mathbf{v}'])$, such that a real MDDH challenge $[\mathbf{v}] = [\mathbf{A}_0\mathbf{r}]$ is transformed into $[\mathbf{v}'] = [\mathbf{A}_1\mathbf{r}]$, and a uniform $[\mathbf{v}]$ is transformed into a uniform $[\mathbf{v}']$. This is obtained simply by picking $\mathbf{U} \leftarrow_R \mathbb{Z}_p^{k \times k}$ and defining $[\mathbf{A}_1]$ as $\overline{[\mathbf{A}_1]} := \overline{[\mathbf{A}_0]}$, $\underline{[\mathbf{A}_1]} := \mathbf{U}\underline{[\mathbf{A}_0]}$, $\overline{[\mathbf{v}']} := \overline{[\mathbf{v}]}$, and $\underline{[\mathbf{v}']} := \mathbf{U}\underline{[\mathbf{v}]}$. With probability $1 - k \cdot 2^{-\Omega(\lambda)}$ over the choices of $\mathbf{A}_0 \leftarrow_R \mathcal{U}_{2k,k}$, $\underline{\mathbf{A}}_0$ is full rank, and $\mathbf{U}\underline{\mathbf{A}}_0$ is uniformly random over $\mathbb{Z}_p^{k \times k}$.

Given $([\mathbf{A}_0], [\mathbf{v}])$, we can compute the tag $\tau := \mathsf{H}(\overline{[\mathbf{v}]})$ and, depending on τ_{i+1}, decide whether we have to switch to $([\mathbf{A}_1], [\mathbf{v}'])$. Note that this does not affect the tag, as it only depends on $\overline{[\mathbf{v}]}$. Now applying the $Q_{\mathbf{enc}}$-fold $\mathcal{U}_{2k,k}$-MDDH a second time allows to change to challenge ciphertexts of the form $[\mathbf{A}_{\tau_{i+1}}\mathbf{r}]$ as desired. Further note that simulating $\mathcal{O}_{\mathbf{dec}}$ only requires knowing $\mathbf{A}^\perp$, which is independent of $\mathbf{A}_0$ and $\mathbf{A}_1$, and therefore, does not compromise the $\mathcal{U}_{2k,k}$-MDDH assumption with respect to those matrices. Using Lemma 4 (random self-reducibility of the $Q_{\mathbf{enc}}$-fold $\mathcal{U}_{2k,k}$-MDDH assumption) and Lemma 3 ($\mathcal{D}_{2k,k}$-MDDH $\Rightarrow$ $\mathcal{U}_{2k,k}$-MDDH), we obtain an adversary $\mathcal{B}$ such that $T(\mathcal{B}) \approx T(\mathcal{A}) + (Q_{\mathbf{enc}} + Q_{\mathbf{dec}}) \cdot \mathsf{poly}(\lambda)$ for a polynomial poly independent of $T(\mathcal{A})$, and such that

$$|\varepsilon_{\mathbf{H}.4.i.0} - \varepsilon_{\mathbf{H}.4.i.1}| \leq 2 \cdot \mathrm{Adv}^{\mathrm{mddh}}_{\mathbb{G},\mathcal{D}_{2k,k},\mathcal{B}}(\lambda) + \frac{2}{p-1}.$$

$\mathbf{H_{4.\mathit{i}.1}} \rightsquigarrow \mathbf{H_{4.\mathit{i}.2}}$**:** For $i = 0, \ldots, \lambda - 1$, the change introduced in $\mathbf{H}_{4.i.2}$ is that $\mathcal{O}_{\mathbf{dec}}(\mathsf{pred}, ([\mathbf{c}], \mathit{\Pi}))$ checks whether $[\mathbf{c}] \in \widetilde{\mathcal{L}_{\mathrm{snd}}}$ (note that this can be checked efficiently given $\mathbf{A}_0$ and $\mathbf{A}_1$). If this is the case, $\mathcal{O}_{\mathbf{dec}}$ continues as in $\mathbf{H}_{4.i.1}$, otherwise, it returns $\perp$. This change can only be detected if the adversary $\mathcal{A}$ manages to submit a valid decryption query with $[\mathbf{c}] \notin \widetilde{\mathcal{L}_{\mathrm{snd}}}$. We bound this event by constructing an adversary $\mathcal{B}$ from $\mathcal{A}$ attacking the constrained $\widetilde{\mathcal{L}_{\mathrm{snd}}}$-soundness of $\widetilde{\mathbf{PS}}$.

On receiving the public parameters ppk of the proof system, $\mathcal{B}$ chooses $\mathbf{k}_0, \mathbf{k}_1 \leftarrow \mathbb{Z}_p^{2k}$ and sends the public key $pk := (ppk, [\mathbf{k}_0^\top \mathbf{A}], [\mathbf{k}_1^\top \mathbf{A}])$ to $\mathcal{A}$.

For answering encryption queries of $\mathcal{A}$, the adversary $\mathcal{B}$ first employs its simulation oracle to obtain $([\mathbf{c}], \mathit{\Pi}, [\kappa])$. Recall that $\mathcal{O}_{\underline{\mathbf{sim}}}$ of $\mathbf{PS}$ returns challenges with $[\mathbf{c}] \in \mathrm{span}([\mathbf{A}_0]) \cup \mathrm{span}([\mathbf{A}_1])$. The adversary then computes $\tau := \mathsf{H}(\overline{[\mathbf{c}]})$ and if $[\mathbf{c}] \notin \mathrm{span}([\mathbf{A}_{\tau_{i+1}}])$ it rejects and queries the simulation oracle again. As $\overline{[\mathbf{A}_0]} = \overline{[\mathbf{A}_1]}$, τ_{i+1} is independent of the span in which $[\mathbf{c}]$ lies. Therefore $\mathcal{B}$ rejects with probability merely $1/2$ and thus requires only $\mathsf{poly}(\lambda) \in O(\lambda)$ time to obtain a query of the desired form with probability $2^{-\Omega(\lambda)}$ (otherwise it aborts), where poly is a polynomial independent of $T(\mathcal{A})$. Finally $\mathcal{B}$ sets $C := ([\mathbf{c}], \mathit{\Pi})$ and $K := (\mathbf{k}_0 + \tau \mathbf{k}_1 + \mathbf{F}_i(\tau_{|i}))^\top [\mathbf{c}] + [\kappa]$ and returns (C, K) to $\mathcal{A}$.

To answer a decryption query $(\mathsf{pred}, ([\mathbf{c}], \mathit{\Pi}))$ the adversary $\mathcal{B}$ has to query its verification oracle for each distinct value $\mathbf{F}_i(\tau_{|i}^{(j)})$, where $\tau^{(j)} \in \mathcal{Q}_{\mathbf{enc}}$, until the simulation oracle replies something other than $\perp$. Note that $\mathbf{F}_i$ can take at most 2^i values, so for small i the number of simulation queries will be much less than $Q_{\mathbf{enc}}$ in general. Nevertheless to keep the bound simpler, we will bound the total running time of the adversary $\mathcal{B}$ to answer decryption queries by $Q_{\mathbf{dec}} \cdot Q_{\mathbf{enc}} \cdot \mathsf{poly}(\lambda)$, where poly is a polynomial independent of $T(\mathcal{A})$.

Namely, on a decryption query $(\mathsf{pred}, ([\mathbf{c}], \mathit{\Pi}))$, the adversary $\mathcal{B}$ computes the tag $\tau^\star := \mathsf{H}(\overline{[\mathbf{c}]})$ as usual and defines for all $\tau^{(j)} \in \mathcal{Q}_{\mathbf{enc}}$ with distinct images $\mathbf{F}_i(\tau_{|i}^{(j)})$ additional predicates $\mathsf{pred}_j \colon \mathbb{G} \to \{0,1\}, K \mapsto \mathsf{pred}\left((\mathbf{k}_0 + \tau^\star \mathbf{k}_1 + \mathbf{F}_i(\tau_{|i}^{(j)}))^\top [\mathbf{c}] + K\right)$. Then for each $j \in [|\mathcal{Q}_{\mathbf{enc}}|]$ adversary $\mathcal{B}$ queries $([\mathbf{c}], \mathit{\Pi}, \mathsf{pred}_j)$ to its verification oracle $\mathcal{O}_{\mathbf{ver}}$, and does the following.

In case $[\mathbf{c}] \in \mathrm{span}([\mathbf{A}])$, the oracle $\mathcal{O}_{\mathbf{ver}}$ returns either $\perp$ or a key $[\kappa]$. In the former case $\mathcal{B}$ forwards $\perp$ to $\mathcal{A}$, in the latter the key $K := (\mathbf{k}_0 + \tau^\star \mathbf{k}_1)^\top [\mathbf{c}] + [\kappa]$.

In case $[\mathbf{c}] \in \mathrm{span}([\mathbf{A}_0]) \cup \mathrm{span}([\mathbf{A}_1])$, $\mathcal{O}_{\mathbf{ver}}$ either returns $\perp$, or the adversary $\mathcal{B}$ loses the constrained soundness game. In case $\mathcal{B}$ has not lost, it forwards $\perp$ to $\mathcal{A}$. Otherwise $\mathcal{A}$ managed to submit a critical query in respect to both games $\mathbf{H}_{4.i.1}$ and $\mathbf{H}_{4.i.2}$ and did thus not succeed in distinguishing between the two.

Finally, in case $[\mathbf{c}] \notin \widetilde{\mathcal{L}_{\mathrm{snd}}}$, $\mathcal{O}_{\mathbf{ver}}$ either returns $\perp$, which $\mathcal{B}$ forwards to $\mathcal{A}$, or it returns "win" to $\mathcal{B}$. Note that only in this case $\mathcal{A}$ managed to submit a valid query outside $\mathcal{L}_{\mathrm{snd}}$ and therefore managed to distinguish between the two games.

Altogether we obtain an adversary $\mathcal{B}$ breaking $\widetilde{\mathcal{L}_{\mathrm{snd}}}$-constrained soundness in time $T(\mathcal{B}) \approx T(\mathcal{A}) + (Q_{\mathbf{enc}} + Q_{\mathbf{enc}} \cdot Q_{\mathbf{dec}}) \cdot \mathsf{poly}(\lambda)$, where poly is a polynomial independent of $T(\mathcal{A})$, such that

$$|\varepsilon_{\mathbf{H}.4.i.1} - \varepsilon_{\mathbf{H}.4.i.2}| \leq \mathrm{Adv}^{\mathrm{csnd}}_{\widetilde{\mathcal{L}_{\mathrm{snd}}}, \widetilde{\mathbf{PS}}, \mathcal{B}}(\lambda) + Q_{\mathbf{enc}} \cdot 2^{-\Omega(\lambda)}.$$

$\mathbf{H_{4.\mathit{i}.2}} \rightsquigarrow \mathbf{H_{4.\mathit{i}.3}}$**:** As described in Fig. 12, game $\mathbf{H}_{4.i.3}$, the oracle $\mathcal{O}_{\mathbf{enc}}$ computes the key using an additional summand $\mathbf{k}_\Delta^{\mathrm{enc}}(\tau)$ for $\tau := \mathsf{H}(\overline{[\mathbf{c}]})$. Similarly, $\mathcal{O}_{\mathbf{dec}}$ uses a vector $\mathbf{k}_\Delta^{\mathrm{dec}}(\tau, [\mathbf{c}])$ for $\tau \in \mathcal{Q}_{\mathbf{enc}}$.

$\underline{\mathcal{O}_{\mathbf{enc}}:}$
$\mathbf{r} \leftarrow_R \mathbb{Z}_p^k$
$[\mathbf{c}] := [\overline{\mathbf{A}_0}]\mathbf{r}$
$\tau := \mathsf{H}(\overline{[\mathbf{c}]})$
$(\Pi, [\kappa]) := \widetilde{\mathbf{PSim}}(ppk, psk, [\mathbf{c}])$
$C := ([\mathbf{c}], \Pi)$
$K := \left(\mathbf{k}_0 + \tau\mathbf{k}_1 + \mathbf{k}_\Delta^{\mathrm{enc}}(\tau)\right)^\top [\mathbf{c}] + [\kappa]$
$\mathcal{C}_{\mathbf{enc}} := \mathcal{C}_{\mathbf{enc}} \cup \{C\}$
return (C, K)

$\underline{\mathcal{O}_{\mathbf{dec}}(\mathsf{pred}, ([\mathbf{c}], \Pi)):}$
$(v, [\kappa]) := \widetilde{\mathbf{PVer}}(psk, [\mathbf{c}], \Pi)$
$\tau^\star := \mathsf{H}(\overline{[\mathbf{c}]})$
if $([\mathbf{c}], \Pi) \notin \mathcal{C}_{\mathbf{enc}}$ **and** $v = 1$ **and** τ is fresh
  if $[\mathbf{c}] \in \mathrm{span}([\mathbf{A}])$
    $K := (\mathbf{k}_0 + \tau^\star\mathbf{k}_1)^\top [\mathbf{c}] + [\kappa]$
    if $\mathsf{pred}(K) = 1$
      return K
  else if $[\mathbf{c}] \in \mathrm{span}([\mathbf{A}_0]) \cup \mathrm{span}([\mathbf{A}_1])$
    $\mathcal{S}_K := \left\{ \left(\mathbf{k}_0 + \tau^\star\mathbf{k}_1 + \mathbf{k}_\Delta^{\mathrm{dec}}(\tau, [\mathbf{c}])\right)^\top [\mathbf{c}] + [\kappa] \,\middle|\, \tau \in \mathcal{Q}_{\mathbf{enc}} \right\}$
    if $\exists K \in \mathcal{S}_K$ **such that** $\mathsf{pred}(K) = 1$
      return *critical query* **and abort**
return $\perp$

Fig. 12: Oracles in Game $\mathbf{H}_{4.i.3}$

In encryption queries $\mathbf{k}_\Delta^{\mathrm{enc}}(\tau)$ for $\tau := \mathsf{H}(\overline{[\mathbf{c}]})$ is defined as

$$\mathbf{k}_\Delta^{\mathrm{enc}}(\tau) := \begin{cases} \mathbf{A}_0^\perp \widetilde{\mathbf{F}}_i^{(0)}(\tau_{|i}) + \mathbf{A}_1^\perp \mathbf{F}_i^{(1)}(\tau_{|i}), & \text{if } \tau_{i+1} = 0 \\ \mathbf{A}_0^\perp \mathbf{F}_i^{(0)}(\tau_{|i}) + \mathbf{A}_1^\perp \widetilde{\mathbf{F}}_i^{(1)}(\tau_{|i}), & \text{if } \tau_{i+1} = 1, \end{cases}$$

where $\mathbf{A}_0^\perp \in \mathsf{orth}([\mathbf{A}_0])$, $\mathbf{A}_1^\perp \in \mathsf{orth}([\mathbf{A}_1])$ and $\mathbf{F}_i^{(0)}, \mathbf{F}_i^{(1)}, \widetilde{\mathbf{F}}_i^{(0)}, \widetilde{\mathbf{F}}_i^{(1)} \colon \{0,1\}^i \to \mathbb{Z}_p^k$ are independent random functions, such that $\mathbf{F}_i(\tau_{|i}) = \mathbf{A}_0^\perp \mathbf{F}_i^{(0)}(\tau_{|i}) + \mathbf{A}_1^\perp \mathbf{F}_i^{(1)}(\tau_{|i})$. Note that with probability $1 - 2^{-\Omega(\lambda)}$ over the choices of $\mathbf{A}_0, \mathbf{A}_1$ the column vectors of $\mathbf{A}_0^\perp$ and $\mathbf{A}_1^\perp$ form a basis of $\mathbb{Z}_p^{2k}$ and thus such $\mathbf{F}_i^{(0)}$, $\mathbf{F}_i^{(1)}$ exist. Further for any bit $b \in \{0,1\}$, and $\mathbf{c} \in \mathrm{span}(\mathbf{A}_b)$ we have

$$\mathbf{k}_\Delta^{\mathrm{enc}}(\tau)^\top \mathbf{c} = \left(\mathbf{k}_\Delta^{\mathrm{enc}}(\tau) + \mathbf{A}_b^\perp \widetilde{\mathbf{F}}_i^{(b)}\right)^\top \mathbf{c}.$$

Thus the change of the encryption oracle is merely conceptional.

The same holds true for the decryption oracle, where we compute the set of admissible keys depending on $[\mathbf{c}]$. Namely, for each tag $\tau \in \mathcal{Q}_{\mathbf{enc}}$, we define $\mathbf{k}_\Delta^{\mathrm{dec}}(\tau, [\mathbf{c}])$ as

$$\mathbf{k}_{\Delta}^{\mathrm{dec}}(\tau,[\mathbf{c}]) := \begin{cases} \mathbf{A}_0^{\perp}\widetilde{\mathbf{F}}_i^{(0)}(\tau_{|i}) + \mathbf{A}_1^{\perp}\mathbf{F}_i^{(1)}(\tau_{|i}), & \text{if } [\mathbf{c}] \in \mathrm{span}([\mathbf{A}_0]) \\ \mathbf{A}_0^{\perp}\mathbf{F}_i^{(0)}(\tau_{|i}) + \mathbf{A}_1^{\perp}\widetilde{\mathbf{F}}_i^{(1)}(\tau_{|i}), & \text{if } [\mathbf{c}] \in \mathrm{span}([\mathbf{A}_1]) \end{cases}$$

Therefore, $\mathbf{H}_{4.i.2}$ and $\mathbf{H}_{4.i.3}$ are identically distributed and we obtain

$$\varepsilon_{\mathbf{H}.4.i.2} = \varepsilon_{\mathbf{H}.4.i.3}.$$

$\mathbf{H_{4.\boldsymbol{i}.3}} \rightsquigarrow \mathbf{H_{4.\boldsymbol{i}.4}}$**:** In game $\mathbf{H}_{4.i.4}$, for $i = 0, \ldots, \lambda - 1$ we define

$$\mathbf{F}_{i+1} \colon \{0,1\}^{i+1} \to \mathbb{Z}_p^{2k}$$

as

$$\mathbf{F}_{i+1}(\tau_{|i+1}) := \begin{cases} \mathbf{A}_0^{\perp}\widetilde{\mathbf{F}}_i^{(0)}(\tau_{|i}) + \mathbf{A}_1^{\perp}\mathbf{F}_i^{(1)}(\tau_{|i}), & \text{if } \tau_{i+1} = 0 \\ \mathbf{A}_0^{\perp}\mathbf{F}_i^{(0)}(\tau_{|i}) + \mathbf{A}_1^{\perp}\widetilde{\mathbf{F}}_i^{(1)}(\tau_{|i}), & \text{if } \tau_{i+1} = 1. \end{cases}$$

Note that this defines a random function, when $\mathbf{F}_i^{(0)}, \mathbf{F}_i^{(1)}, \widetilde{\mathbf{F}}_i^{(0)}, \widetilde{\mathbf{F}}_i^{(1)} : \{0,1\}^i \to \mathbb{Z}_p^k$ are independent random functions.

Similarly, in decryption queries for $\tau \in \mathcal{Q}_{\mathbf{dec}}$ we use $\mathbf{F}_{i+1}$ as defined above applied to $\tau_{|i} d_{[\mathbf{c}]}$, where $d_{[\mathbf{c}]}$ is defined as

$$d_{[\mathbf{c}]} := \begin{cases} 0, & \text{if } [\mathbf{c}] \in \mathrm{span}([\mathbf{A}_0]) \\ 1, & \text{if } [\mathbf{c}] \in \mathrm{span}([\mathbf{A}_1]). \end{cases}$$

As the changes again are merely conceptional, we have

$$\varepsilon_{\mathbf{H}.4.i.3} = \varepsilon_{\mathbf{H}.4.i.4}.$$

$\mathbf{H_{4.\boldsymbol{i}.4}} \rightsquigarrow \mathbf{H_{4.\boldsymbol{i}.5}}$**:** From game $\mathbf{H}_{4.i.5}$ on, we again allow decryption queries outside $\widetilde{\mathcal{L}_{\mathrm{snd}}}$. This can only increase the winning chances of the adversary, because it does not change the view on non-critical queries. We thus have

$$\varepsilon_{\mathbf{H}.4.i.4} \leq \varepsilon_{\mathbf{H}.4.i.5}.$$

$\mathbf{H_{4.\boldsymbol{i}.5}} \rightsquigarrow \mathbf{H_{4.\boldsymbol{i}.6}}$**:** Game $\mathbf{H}_{4.i.6}$, for $i = 0, \ldots, \lambda - 1$, is identical to $\mathbf{H}_{4.i.5}$, except for $\mathcal{O}_{\mathbf{dec}}$, which now computes the set of valid keys as

$$\mathcal{S}_K := \left\{ \left(\mathbf{k}_0 + \tau^{\star}\mathbf{k}_1 + \mathbf{F}_{i+1}(\tau_{|i}\, b) \right)^{\top} [\mathbf{c}] \;\middle|\; \tau \in \mathcal{Q}_{\mathbf{enc}}, b \in \{0,1\} \right\}$$

Note that this set includes the set of keys computed in $\mathbf{H}_{4.i.5}$. Therefore, this increases the probability of the adversary to submit a critical query, while not changing its view on non-critical queries. In conclusion,

$$\varepsilon_{\mathbf{H}.4.i.5} \leq \varepsilon_{\mathbf{H}.4.i.6}.$$

$\mathbf{H_{4.\boldsymbol{i}.6}} \rightsquigarrow \mathbf{H_{4.\boldsymbol{i}.7}}$**:** Game $\mathbf{H}_{4.i.7}$, for $i = 0, \ldots, \lambda - 1$, is identical to $\mathbf{H}_{4.i.6}$, except for $\mathcal{O}_{\mathbf{dec}}$, which now computes the set of valid keys as

$$\mathcal{S}_K := \left\{ \left(\mathbf{k}_0 + \tau^\star \mathbf{k}_1 + \mathbf{F}_{i+1}(\tau_{|i}\ \tau_{i+1})\right)^\top [\mathbf{c}] \,\middle|\, \tau \in \mathcal{Q}_{\mathbf{enc}}, \right\}.$$

It suffices to show that with all but negligible probability, there is no key in $\mathcal{S}_K$ which corresponds to a tag $\tau \in \mathcal{Q}_{\mathbf{enc}}$ and a bit $b \in \{0,1\}$ such that $\tau_{|i}b \in \{0,1\}^{i+1}$ is not the prefix of any tag in $\mathcal{Q}_{\mathbf{enc}}$, and that satisfies pred. We proceed via a hybrid argument over all queries to $\mathcal{O}_{\mathbf{dec}}$. To that end, we introduce intermediate games $\mathbf{H}_{4.i.6.j}$ for $j = 0, \ldots, Q_{\mathbf{dec}}$, defined as $\mathbf{H}_{4.i.6}$, except that $\mathcal{O}_{\mathbf{dec}}$ proceeds as in game $\mathbf{H}_{4.i.7}$ on its j-th last queries. We show that:

$$\mathbf{H}_{4.i.6} = \mathbf{H}_{4.i.6.0} \approx_s \mathbf{H}_{4.i.6.1} \approx_s \ldots \approx_s \mathbf{H}_{4.i.6.Q_{\mathbf{dec}}} = \mathbf{H}_{4.i.7},$$

where by $\approx_s$ we denote statistical closeness. We show that for all $j = 0, \ldots, Q_{\mathbf{dec}} - 1$,

$$|\varepsilon_{\mathbf{H}.4.i.6.j} - \varepsilon_{\mathbf{H}.4.i.6.j+1}| \leq Q_{\mathbf{enc}} \cdot \Pr_{K \leftarrow_R \mathcal{K}}[\mathsf{pred}_{j+1}(K) = 1].$$

This is because for all tags $\tau \in \mathcal{Q}_{\mathbf{enc}}$ and $b \in \{0,1\}$ such that $\tau_{|i}b \in \{0,1\}^{i+1}$ is not prefix of any $\tau \in \mathcal{Q}_{\mathbf{enc}}$, the value $\mathbf{F}_{i+1}(\tau_{|i}b)$ is a random value, uniform over $\mathbb{Z}_p^k$, independent of $\mathcal{A}$'s view before its $(j+1)$-st query to $\mathcal{O}_{\mathbf{dec}}$. Summing up, we obtain

$$|\varepsilon_{\mathbf{H}.4.i.6} - \varepsilon_{\mathbf{H}.4.i.7}| \leq Q_{\mathbf{enc}} \cdot Q_{\mathbf{dec}} \cdot \mathrm{uncert}_{\mathcal{A}}(\lambda).$$

$\mathbf{H_{4.\boldsymbol{i}.7}} \rightsquigarrow \mathbf{H_{4.(\boldsymbol{i}+1).0}}$**:** For $i = 0, \ldots, \lambda - 1$, in $\mathbf{H}_{4.(i+1).0}$ the challenge ciphertexts are switched back to the span of $[\mathbf{A}_0]$ independent of the tag τ, the transition is thus the reverse to $\mathbf{H}_{4.i.0} \rightsquigarrow \mathbf{H}_{4.i.1}$. More precisely, we first tightly switch all challenges of the form $[\mathbf{A}_{\tau_{i+1}}\mathbf{r}]$ to uniform random vectors over $\mathbb{G}^{2k}$ and then back to vectors in the span of $[\mathbf{A}_0]$. From an adversary $\mathcal{A}$ detecting this change, we can construct an adversary $\mathcal{B}$ attacking the $Q_{\mathbf{enc}}$-fold $\mathcal{U}_{2k,k}$-MDDH assumption as follows. On input $([\mathbf{A}_0], [\mathbf{v}_1| \cdots |\mathbf{v}_{Q_{\mathbf{enc}}}])$ with $[\mathbf{A}_0] \in \mathbb{G}^{2k \times k}$ and $[\mathbf{V}] := [\mathbf{v}_1| \cdots |\mathbf{v}_{Q_{\mathbf{enc}}}] \in \mathbb{G}^{2k \times Q_{\mathbf{enc}}}$, the adversary $\mathcal{B}$ chooses $\mathbf{U} \leftarrow \mathbb{Z}_p^{k \times k}$ and sets $[\mathbf{A}_1]$ such that $\overline{[\mathbf{A}_1]} = \overline{[\mathbf{A}_0]}$ and $\underline{[\mathbf{A}_1]} = \mathbf{U}\underline{[\mathbf{A}_0]}$. With probability $1 - k \cdot 2^{-\Omega(\lambda)}$ over the choices of $\mathbf{A}_0 \leftarrow_R \mathcal{U}_{2k,k}$, $\underline{\mathbf{A}}_0$ is full rank, and $\mathbf{U}\underline{\mathbf{A}}_0$ is uniformly random over $\mathbb{Z}_p^{k \times k}$.

Further $\mathcal{B}$ chooses the rest of the public parameters as in Section 4.1 and generates the public and secret keys of the KEM by invoking $\mathbf{KGen}$ on input 1^λ. On the j-th query of $\mathcal{A}$ to $\mathcal{O}_{\mathbf{enc}}$, $\mathcal{B}$ computes $\tau := \mathsf{H}(\overline{[\mathbf{v}_j]})$. In case $\tau_{i+1} = 0$, the adversary continues answering the decryption query with $[\mathbf{c}] := [\mathbf{v}_j]$. In case $\tau_{i+1} = 1$, the adversary instead sets $[\mathbf{c}]$ such that $\overline{[\mathbf{c}]} = \overline{[\mathbf{v}_j]}$ and $\underline{[\mathbf{c}]} = \mathbf{U}\underline{[\mathbf{v}_j]}$. In case $[\mathbf{V}]$ was uniformly random over $\mathbb{G}^{2k \times Q_{\mathbf{enc}}}$, the adversary $\mathcal{B}$ simulates the intermediary game, where all challenge ciphertexts are chosen uniformly random. If instead for each $j \in \{1, \ldots, Q_{\mathbf{enc}}\}$ there exists an $\mathbf{r}_j \in \mathbb{Z}_p^k$ such that $[\mathbf{v}_j] = [\mathbf{A}_0]\mathbf{r}_j$, the adversary simulates game $\mathbf{H}_{4.i.7}$, as in this case for all $j \in \{1, \ldots, Q_{\mathbf{enc}}\}$ we have $[\mathbf{c}_j] = [\mathbf{A}_{\tau_{i+1}}\mathbf{r}_j]$.

Now we can employ the $Q_{\mathbf{enc}}$-fold $\mathcal{U}_{2k,k}$-MDDH assumption a second time to tightly switch back the challenge ciphertexts from random to the span of $[\mathbf{A}_0]$.

Finally, using Lemma 4 (random self-reducibility of the $\mathcal{U}_{2k,k}$-MDDH assumption) and Lemma 3 ($\mathcal{D}_{2k,k}$-MDDH $\Rightarrow \mathcal{U}_{2k,k}$-MDDH), we obtain an adversary $\mathcal{B}'$ such that $T(\mathcal{B}') \approx T(\mathcal{A}) + (Q_{\mathbf{enc}} + Q_{\mathbf{dec}}) \cdot \mathsf{poly}(\lambda)$ for a polynomial poly independent of $T(\mathcal{A})$, and such that

$$|\varepsilon_{\mathbf{H}.4.i.5} - \varepsilon_{\mathbf{H}.4.(i+1).0}| \leq 2 \cdot \mathrm{Adv}^{\mathrm{mddh}}_{\mathbb{G},\mathcal{D}_{2k,k},\mathcal{B}'}(\lambda) + \frac{2}{p-1}.$$

Game $\mathbf{H_5}$: We now show that an adversary has only negligible chances to win $\mathbf{H}_5 := \mathbf{H}_{4.\lambda.7}$. We argue as follows.

First, for $\mathbf{u} \leftarrow_R \mathbb{Z}_p^k$ the tuples

$$\left(\mathbf{k}_1, \left(\mathbf{F}_\lambda(\tau)\right)_{\tau \in \{0,1\}^\lambda}\right) \text{ and } \left(\mathbf{k}_1 - \mathbf{A}^\perp \mathbf{u}, \left(\mathbf{F}_\lambda(\tau) + \tau \mathbf{A}^\perp \mathbf{u}\right)_{\tau \in \{0,1\}^\lambda}\right)$$

are distributed identically.

Second, the set of tags computed by $\mathcal{O}_{\mathbf{enc}}$ and the set of tags computed by $\mathcal{O}_{\mathbf{dec}}$ are disjoint (recall that we established this in game $\mathbf{G}_1$ in the proof of Theorem 3).

Note that $\mathbf{u}$ does not show up when $\mathcal{O}_{\mathbf{enc}}$ computes challenge keys, since in this case

$$\begin{aligned} K &= \left(\mathbf{k}_0 + \tau\left(\mathbf{k}_1 - \mathbf{A}^\perp \mathbf{u}\right) + \mathbf{F}_\lambda(\tau) + \tau \mathbf{A}^\perp \mathbf{u}\right)^\top [\mathbf{c}] \\ &= \left(\mathbf{k}_0 + \tau \mathbf{k}_1 + \mathbf{F}_\lambda(\tau)\right)^\top [\mathbf{c}], \end{aligned}$$

that is, the extra terms cancel each other out.

On the contrary, an extra term appears when $\mathcal{O}_{\mathbf{dec}}$ is queried on an input that contains $[\mathbf{c}]$ such that $\mathbf{c}^\top \mathbf{A}^\perp \neq 0$, since $\mathcal{O}_{\mathbf{dec}}$ computes $\tau^\star := \mathsf{H}(\overline{[\mathbf{c}]})$ and the set of keys as

$$\mathcal{S}_K := \left\{ \left(\mathbf{k}_0 + \tau^\star \mathbf{k}_1 + \mathbf{F}_\lambda(\tau) + (\tau^\star - \tau)\mathbf{A}^\perp \mathbf{u}\right)^\top [\mathbf{c}] \;\middle|\; \tau \in \mathcal{Q}_{\mathbf{enc}} \right\}.$$

As we require tags to be fresh, we have $\tau^\star \notin \mathcal{Q}_{\mathbf{enc}}$ and therefore the term $(\tau^\star - \tau)(\mathbf{A}^\perp \mathbf{u})^\top \mathbf{c}$ is uniformly random over $\mathbb{Z}_p$. Thus, the marginal distribution of each key in $\mathcal{S}_K$ is uniform over $\mathbb{G}$. Using a hybrid argument over all queries to $\mathcal{O}_{\mathbf{dec}}$, we hence obtain

$$|\varepsilon_{\mathbf{H}.5}| \leq Q_{\mathbf{dec}} \cdot Q_{\mathbf{enc}} \cdot \mathrm{uncert}_\mathcal{A}(\lambda).$$

$\square$

References

[1] Masayuki Abe, Bernardo David, Markulf Kohlweiss, Ryo Nishimaki, and Miyako Ohkubo. "Tagged One-Time Signatures: Tight Security and Optimal Tag Size". In: *PKC 2013*. Ed. by Kaoru Kurosawa and Goichiro Hanaoka. Vol. 7778. LNCS. Springer, Heidelberg, 2013, pp. 312–331. DOI: 10.1007/978-3-642-36362-7_20.

[2] Nuttapong Attrapadung, Goichiro Hanaoka, and Shota Yamada. "A Framework for Identity-Based Encryption with Almost Tight Security". In: *ASIACRYPT 2015, Part I*. Ed. by Tetsu Iwata and Jung Hee Cheon. Vol. 9452. LNCS. Springer, Heidelberg, 2015, pp. 521–549. DOI: 10.1007/978-3-662-48797-6_22.

[3] Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. "Public-Key Encryption in a Multi-user Setting: Security Proofs and Improvements". In: *EUROCRYPT 2000*. Ed. by Bart Preneel. Vol. 1807. LNCS. Springer, Heidelberg, May 2000, pp. 259–274.

[4] Olivier Blazy, Eike Kiltz, and Jiaxin Pan. "(Hierarchical) Identity-Based Encryption from Affine Message Authentication". In: *CRYPTO 2014, Part I*. Ed. by Juan A. Garay and Rosario Gennaro. Vol. 8616. LNCS. Springer, Heidelberg, Aug. 2014, pp. 408–425. DOI: 10.1007/978-3-662-44371-2_23.

[5] Jie Chen and Hoeteck Wee. "Fully, (Almost) Tightly Secure IBE and Dual System Groups". In: *CRYPTO 2013, Part II*. Ed. by Ran Canetti and Juan A. Garay. Vol. 8043. LNCS. Springer, Heidelberg, Aug. 2013, pp. 435–460. DOI: 10.1007/978-3-642-40084-1_25.

[6] Ronald Cramer and Victor Shoup. "Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack". In: *SIAM Journal on Computing* 33.1 (2003), pp. 167–226.

[7] Ronald Cramer and Victor Shoup. "Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption". In: *EUROCRYPT 2002*. Ed. by Lars R. Knudsen. Vol. 2332. LNCS. Springer, Heidelberg, 2002, pp. 45–64.

[8] Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Villar. “An Algebraic Framework for Diffie-Hellman Assumptions”. In: *CRYPTO 2013, Part II*. Ed. by Ran Canetti and Juan A. Garay. Vol. 8043. LNCS. Springer, Heidelberg, Aug. 2013, pp. 129–147. DOI: 10.1007/978-3-642-40084-1_8.

[9] Romain Gay, Dennis Hofheinz, Eike Kiltz, and Hoeteck Wee. “Tightly CCA-Secure Encryption Without Pairings”. In: *EUROCRYPT 2016, Part I*. Ed. by Marc Fischlin and Jean-Sébastien Coron. Vol. 9665. LNCS. Springer, Heidelberg, May 2016, pp. 1–27. DOI: 10.1007/978-3-662-49890-3_1.

[10] Junqing Gong, Jie Chen, Xiaolei Dong, Zhenfu Cao, and Shaohua Tang. “Extended Nested Dual System Groups, Revisited”. In: *PKC 2016, Part I*. Ed. by Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang. Vol. 9614. LNCS. Springer, Heidelberg, Mar. 2016, pp. 133–163. DOI: 10.1007/978-3-662-49384-7_6.

[11] Dennis Hofheinz. “Adaptive Partitioning”. In: *Eurocrypt 2017 (to appear)*. 2017.

[12] Dennis Hofheinz. “Algebraic Partitioning: Fully Compact and (almost) Tightly Secure Cryptography”. In: *TCC 2016-A, Part I*. Ed. by Eyal Kushilevitz and Tal Malkin. Vol. 9562. LNCS. Springer, Heidelberg, Jan. 2016, pp. 251–281. DOI: 10.1007/978-3-662-49096-9_11.

[13] Dennis Hofheinz and Tibor Jager. “Tightly Secure Signatures and Public-Key Encryption”. In: *CRYPTO 2012*. Ed. by Reihaneh Safavi-Naini and Ran Canetti. Vol. 7417. LNCS. Springer, Heidelberg, Aug. 2012, pp. 590–607.

[14] Dennis Hofheinz and Eike Kiltz. “Secure Hybrid Encryption from Weakened Key Encapsulation”. In: *CRYPTO 2007*. Ed. by Alfred Menezes. Vol. 4622. LNCS. Springer, Heidelberg, Aug. 2007, pp. 553–571.

[15] Dennis Hofheinz, Jessica Koch, and Christoph Striecks. “Identity-Based Encryption with (Almost) Tight Security in the Multi-instance, Multi-ciphertext Setting”. In: *PKC 2015*. Ed. by Jonathan Katz. Vol. 9020. LNCS. Springer, Heidelberg, 2015, pp. 799–822. DOI: 10.1007/978-3-662-46447-2_36.

[16] Russell Impagliazzo, Leonid A. Levin, and Michael Luby. “Pseudo-random Generation from one-way functions (Extended Abstracts)”. In: *21st ACM STOC*. ACM Press, May 1989, pp. 12–24.

[17] Kaoru Kurosawa and Yvo Desmedt. “A New Paradigm of Hybrid Encryption Scheme”. In: *CRYPTO 2004*. Ed. by Matthew Franklin. Vol. 3152. LNCS. Springer, Heidelberg, Aug. 2004, pp. 426–442.

[18] Arjen K. Lenstra and Eric R. Verheul. “Selecting Cryptographic Key Sizes”. In: *Journal of Cryptology* 14.4 (2001), pp. 255–293.

[19] Benoît Libert, Marc Joye, Moti Yung, and Thomas Peters. “Concise Multi-challenge CCA-Secure Encryption and Signatures with Almost Tight Security”. In: *ASIACRYPT 2014, Part II*. Ed. by Palash Sarkar and Tetsu Iwata. Vol. 8874. LNCS. Springer, Heidelberg, Dec. 2014, pp. 1–21. DOI: 10.1007/978-3-662-45608-8_1.

[20] Benoît Libert, Thomas Peters, Marc Joye, and Moti Yung. “Compactly Hiding Linear Spans - Tightly Secure Constant-Size Simulation-Sound QA-NIZK Proofs and Applications”. In: *ASIACRYPT 2015, Part I*. Ed. by Tetsu Iwata and Jung Hee Cheon. Vol. 9452. LNCS. Springer, Heidelberg, 2015, pp. 681–707. DOI: 10.1007/978-3-662-48797-6_28.

A The OR-Proof

Lemma 8 (Constrained $\widetilde{\mathcal{L}_{\text{snd}}}$-soundness of $\widetilde{\mathbf{PS}}$ for $k = 1$). *If the DDH assumption holds in $\mathbb{G}$ and h_0, h_1 are universal hash functions, then the proof system described in Fig. 4 complies with constrained $\widetilde{\mathcal{L}_{\text{snd}}}$-soundness. Namely, for any adversary $\mathcal{A}$ against $\widetilde{\mathcal{L}_{\text{snd}}}$-soundness, there exists an adversary $\mathcal{B}$ such that $T(\mathcal{B}) \approx T(\mathcal{A}) + (Q_{\mathbf{dec}} + Q_{\mathbf{ver}}) \cdot \mathsf{poly}(\lambda)$ and*

$$\begin{aligned}\mathrm{Adv}^{\mathrm{csnd}}_{\widetilde{\mathcal{L}_{\text{snd}}},\widetilde{\mathbf{PS}},\mathcal{A}}(\lambda) \leq \mathrm{Adv}^{\mathrm{mddh}}_{\mathbb{G},\mathcal{B},\mathcal{D}_{2k,k}}(\lambda) &+ Q_{\mathbf{ver}} \cdot \mathrm{uncert}_{\mathcal{A}}(\lambda)\\ &+ (Q_{\mathbf{sim}} + Q_{\mathbf{ver}}) \cdot 2^{-\Omega(\lambda)},\end{aligned}$$

where $Q_{\mathbf{ver}}$, $Q_{\mathbf{dec}}$ are the number of calls to $\mathcal{O}_{\mathbf{ver}}$ and $\mathcal{O}_{\mathbf{dec}}$ respectively, $\mathrm{uncert}^{\mathrm{snd}}_{\mathcal{A}}(\lambda)$ *describes the uncertainty of the predicates provided by $\mathcal{A}$ and* poly *is a polynomial function, independent of $T(\mathcal{A})$.*

Proof. We prove the $\widetilde{\mathcal{L}_{\text{snd}}}$-soundness of $\widetilde{\mathbf{PS}}$ via a series of games, described in Fig. 13. We start by giving a short overview of the proof.

The idea is to first randomize the x that is used in simulated proof of statements $[\mathbf{c}] \in \mathcal{L}_{\text{snd}} \setminus \mathcal{L}$, using the DDH assumption and the Leftover Hash Lemma (Lemma 1). For the final proof step, let $([\mathbf{c}], [\pi], [\kappa])$ be an arbitrary combined proof for $[\mathbf{c}] \in \widetilde{\mathcal{L}_{\text{snd}}}$, that is, such that there exists $r \in \mathbb{Z}_p$ such that either $[\mathbf{c}] = [\mathbf{A}r]$, $[\mathbf{c}] = [\mathbf{A}_0 r]$ or $[\mathbf{c}] = [\mathbf{A}_1 r]$. In the first case, nothing about $[\widetilde{\mathbf{K}}_y]$ is leaked. In case $[\mathbf{c}]$ is in the span of $[\mathbf{A}_0]$, we have $[\pi, \kappa] = [\mathbf{A}_0] \cdot x + [\mathbf{c}] \cdot y = [\mathbf{A}_0](x + r \cdot y)$, thus y, and in particular $\widetilde{\mathbf{K}}_y$, are completely hidden by the randomized x. Finally, in case $[\mathbf{c}] = [\mathbf{A}_1 r]$ we have $y = \mathsf{h}_1([\widetilde{\mathbf{K}}_y \mathbf{A}_1 r])$, and thus informationtheoretically only $[\widetilde{\mathbf{K}}_y \mathbf{A}_1]$ is leaked. This implies (via randomizing $\widetilde{\mathbf{K}}_y$ by adding a term $\mathbf{U}(\mathbf{A}_1^{\perp})^{\top}$ for $\mathbf{U} \leftarrow_R \mathbb{Z}_p^{2\times 1}$ and $\mathbf{A}_1^{\perp} \in \mathsf{orth}(\mathbf{A}_1)$) that even knowing many sound tuples $([\mathbf{c}], [\pi], [\kappa])$ for $[\mathbf{c}] \in \widetilde{\mathcal{L}_{\text{snd}}}$, an adversary cannot do better than guessing y to produce a valid key for statements outside $\widetilde{\mathcal{L}_{\text{snd}}}$, and therefore, only has negligible winning chances.

We start with the $\widetilde{\mathcal{L}_{\text{snd}}}$-constrained soundness game, which we refer to as game $\mathbf{G}$. In the following we want to bound the probability

$$\varepsilon := \mathrm{Adv}^{\mathrm{csnd}}_{\mathbf{PS},\mathcal{A}}(\lambda).$$

We denote the probability that the adversary $\mathcal{A}$ wins the game $\mathbf{G}_i$ by

$$\varepsilon_i := \mathrm{Adv}_{\mathbf{G}_i,\mathcal{A}}(\lambda).$$

We omit the proof of the game transitions, as they almost verbatim follow the proof of Theorem 1, where $\mathcal{L}_{\text{snd}}$ is replaced by $\widetilde{\mathcal{L}_{\text{snd}}}$. It is left to show that the adversary has only negligible chances in winning game $\mathbf{G}_3$.

Game $\mathbf{G}_3$: We show that $\varepsilon_3 \leq Q_{\mathbf{ver}} \cdot \mathrm{uncert}^{\mathrm{snd}}_{\mathcal{A}}(\lambda)$, where $Q_{\mathbf{ver}}$ is the number of queries to $\mathcal{O}_{\mathbf{ver}}$ and $\mathrm{uncert}^{\mathrm{snd}}_{\mathcal{A}}(\lambda)$ describes the uncertainty of the predicates provided by the adversary as described in Definition 12.

We use a hybrid argument over the $Q_{\mathbf{ver}}$ queries to $\mathcal{O}_{\mathbf{ver}}$. To that end, we introduce games $\mathbf{G}_{3.i}$ for $i = 0, \ldots, Q_{\mathbf{ver}}$, defined as $\mathbf{G}_3$ except that for its first i queries $\mathcal{O}_{\mathbf{ver}}$ answers $\perp$ on any input $([\mathbf{c}], [\pi], \mathsf{pred})$ with $[\mathbf{c}] \notin \widetilde{\mathcal{L}_{\text{snd}}}$. We have $\varepsilon_3 = \varepsilon_{3.0}$, $\varepsilon_{3.Q_{\mathbf{ver}}} = 0$ and we show for all $i \in [Q_{\mathbf{ver}}]$ it holds

$$|\varepsilon_{3.i-1} - \varepsilon_{3.i}| \leq \Pr_{K \in \mathcal{K}}[\mathsf{pred}_i(K) = 1] + 2^{-\Omega(\lambda)},$$

#	sim. x for $[\mathbf{c}] \in \widetilde{\mathcal{L}_{\mathrm{snd}}} \backslash \mathcal{L}$	ver. $[\kappa]$ for $[\mathbf{c}] \notin \mathcal{L}$	game knows	remark
$\mathbf{G}_0$	$x := \mathsf{h}_0\left(\mathbf{K}_x[\mathbf{c}]\right)$	$\underline{[\mathbf{A}_0]} \cdot \mathbf{x} + \underline{[\mathbf{c}]} \cdot y$		$\widetilde{\mathcal{L}_{\mathrm{snd}}}$-soundn. game w/o *lose*
$\mathbf{G}_1$	$x := \mathsf{h}_0\left(\mathbf{K}_x[\mathbf{c}]\right)$	$\underline{\mathbf{A}}_0 \overline{\mathbf{A}_0}^{-1} \left([\pi^\star] - \overline{[\mathbf{c}]} \cdot y\right) + \underline{[\mathbf{c}]} \cdot y$	$\mathbf{A}, \mathbf{A}_0$	win. chances increase
$\mathbf{G}_2$	$\mathbf{u} \leftarrow_R \mathbb{Z}_p^2$, $x := \mathsf{h}_0([\mathbf{u}])$	$\underline{\mathbf{A}}_0 \overline{\mathbf{A}_0}^{-1} \left([\pi^\star] - \overline{[\mathbf{c}]} \cdot y\right) + \underline{[\mathbf{c}]} \cdot y$	$\mathbf{A}, \mathbf{A}_0$	DDH
$\mathbf{G}_3$	$x \leftarrow_R \mathbb{Z}_p$	$\underline{\mathbf{A}}_0 \overline{\mathbf{A}_0}^{-1} \left([\pi^\star] - \overline{[\mathbf{c}]} \cdot y\right) + \underline{[\mathbf{c}]} \cdot y$	$\mathbf{A}, \mathbf{A}_0$	Lemma 1 (LOHL)

Fig. 13: Overview of the proof of $\widetilde{\mathcal{L}_{\mathrm{snd}}}$-constrained soundness of $\mathbf{PS}$. The first column shows how x is computed for queries to $\mathcal{O}_{\mathbf{sim}}$. The second column shows how the key $[\kappa]$ computed by the verifier in queries to $\mathcal{O}_{\mathbf{ver}}$ when $[\mathbf{c}] \notin \mathcal{L}$. The third column "game knows" gives an overview of which non-public information need to be known by the game respective to $\mathbf{A}$, $\mathbf{A}_0$ and $\mathbf{A}_1$.

where pred_i is the predicate contained in the i-th query to $\mathcal{O}_{\mathbf{ver}}$.

Games $\mathbf{G}_{3.i}$ and $\mathbf{G}_{3.(i+1)}$ behave identically on the first i queries to $\mathcal{O}_{\mathbf{ver}}$. An adversary can only distinguish between the two, if it manages to provide a valid $(i+1)$-st query $([\mathbf{c}], [\pi], \mathsf{pred})$ to $\mathcal{O}_{\mathbf{ver}}$ with $[\mathbf{c}] \notin \widetilde{\mathcal{L}_{\mathrm{snd}}}$. In the following we bound the probability of this happening.

From queries to $\mathcal{O}_{\mathbf{sim}}$ and the first i queries to $\mathcal{O}_{\mathbf{ver}}$ the adversary can only learn valid tuples $([\mathbf{c}], [\pi], [\kappa])$ with $[\mathbf{c}] \in \widetilde{\mathcal{L}_{\mathrm{snd}}}$. As explained in the beginning, such combined proofs reveal nothing about $\widetilde{\mathbf{K}}_y$ beyond $[\widetilde{\mathbf{K}}_y \mathbf{A}_1]$, as either $\widetilde{\mathbf{K}}_y$ is not employed for the computation of y, $[\mathbf{c}] = [\mathbf{A}_1 r]$ for an $r \in \mathbb{Z}_p$ and $y = \mathsf{h}_1([\widetilde{\mathbf{K}}_y \mathbf{c}]) = \mathsf{h}_1([\widetilde{\mathbf{K}}_y \mathbf{A}]r)$ or $[\mathbf{c}] = [\mathbf{A}_0 r]$ and $[\pi, \kappa] = [\mathbf{A}_0](x + r \cdot y)$. In the latter case y is hidden by the fully randomized x.

For any $[\mathbf{c}] \notin \widetilde{\mathcal{L}_{\mathrm{snd}}}$, $y = \mathsf{h}_1[\widetilde{\mathbf{K}}_y \mathbf{c}]$ computed by $\mathcal{O}_{\mathbf{ver}}$ is distributed statistically close to uniform from the adversary's point of view because of the following. First we can replace $\widetilde{\mathbf{K}}_y$ by $\widetilde{\mathbf{K}}_y + \mathbf{U}(\mathbf{A}_1^\perp)^\top$ for $\mathbf{U} \leftarrow_R \mathbb{Z}_p^{2\times 1}$ and $\mathbf{A}_1^\perp \in \mathsf{orth}(\mathbf{A}_1)$ as both are distributed identically. This extra term is not revealed through the public key, through simulation queries or through the first i oracle queries to $\mathcal{O}_{\mathbf{ver}}$ by the our previous considerations.

Now Lemma 1 (Leftover Hash Lemma) implies that the distribution of y is statistically close to uniform as desired. Since $[\mathbf{c}] \notin \mathrm{span}([\mathbf{A}_0])$ we have $\underline{[\mathbf{c}]} - \underline{[\mathbf{A}_0]}\overline{\mathbf{A}_0}^{-1}\overline{[\mathbf{c}]} \neq 0$, thus also the key

$$[\kappa] := \underline{\mathbf{A}}_0 \overline{\mathbf{A}_0}^{-1} [\pi^\star] + \underbrace{\left(\underline{[\mathbf{c}]} - \underline{\mathbf{A}}_0 \overline{\mathbf{A}_0}^{-1} \overline{[\mathbf{c}]}\right)}_{\neq 0} \cdot y$$

computed by $\mathcal{O}_{\mathbf{ver}}$ is statistically close to uniformly random over $\mathbb{Z}_p$. Altogether this yields

$$|\varepsilon_{3.i} - \varepsilon_{3.(i+1)}| \leq \Pr_{K \in \mathcal{K}}[\mathsf{pred}_i(K) = 1] + 2^{-\Omega(\lambda)}.$$

In conclusion, we obtain

$$\varepsilon_3 \leq Q_{\mathbf{ver}} \cdot \mathrm{uncert}_{\mathcal{A}}(\lambda) + Q_{\mathbf{ver}} \cdot 2^{-\Omega(\lambda)}.$$

$\square$

Theorem 2. *If the $\mathcal{D}_{2k,k}$-MDDH assumption holds in $\mathbb{G}$ and h_0, h_1, and h_2 are universal hash functions, then the proof system* $\mathbf{PS}$ *described in Fig. 5 is $\mathcal{L}_{\mathrm{snd}}$-qualified. Further, the proof system* $\mathbf{PS}$ *is $\widetilde{\mathcal{L}_{\mathrm{snd}}}$-extensible.*

Proof (Sketch). *Completeness* and *perfect zero-knowledge* follow from the according properties of the underlying proof system $Pre\mathbf{PS}$ (Fig. 2), which can be proven analogous to the case $k = 1$. *Uniqueness* of the keys follows from the fact that the verification algorithm computes exactly one proof $[\pi_1, \pi_2]$, and aborts if $[\pi_1, \pi_2] \neq [\pi^\star]$. For the $\widetilde{\mathcal{L}_{\mathrm{snd}}}$-extensibility we refer to Lemma 11 and Lemma 12. We only sketch the proof of constrained $\mathcal{L}_{\mathrm{snd}}$-soundness (see Lemma 10), as it is similar to the proof for the case $k = 1$. We mostly argue about the underlying proof system $Pre\mathbf{PS}$, as this directly translates to $\mathbf{PS}$. We give an overview of the games in Fig. 14. The main difference with the case $k = 1$ is the argument used to prove that in Game $\mathbf{G}_3$, the keys $[\kappa]$ computed by $\mathcal{O}_{\mathbf{ver}}$ is uniformly random over $\mathbb{G}$. $\square$

Lemma 10 (Constrained $\mathcal{L}^{\mathrm{snd}}$-soundness of PS for $k > 1$). *If the $\mathcal{D}_{2k,k}$-MDDH assumption holds in $\mathbb{G}$, and h_0, h_1 are universal hash functions, then the proof system* $\mathbf{PS}$ *described in Fig. 5 (for $k > 1$) complies with constrained $\mathcal{L}^{\mathrm{snd}}$-soundness. More precisely, for every adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that $T(\mathcal{B}) \approx T(\mathcal{A}) + (Q_{\mathbf{sim}} + Q_{\mathbf{ver}}) \cdot \mathsf{poly}(\lambda)$ and*

$$\mathrm{Adv}^{\mathrm{csnd}}_{\mathbf{PS},\mathcal{A}}(\lambda) \leq \mathrm{Adv}^{\mathrm{mddh}}_{\mathbb{G},\mathcal{D}_{2k,k},\mathcal{B}}(\lambda) + (Q_{\mathbf{sim}} + Q_{\mathbf{ver}}) \cdot 2^{-\Omega(\lambda)},$$

where $Q_{\mathbf{ver}}$, $Q_{\mathbf{sim}}$ are the number of calls to $\mathcal{O}_{\mathbf{ver}}$ and $\mathcal{O}_{\mathbf{sim}}$ respectively, $\mathrm{uncert}^{\mathrm{snd}}_{\mathcal{A}}(\lambda)$ *describes the uncertainty of the predicates provided by $\mathcal{A}$ (see Definition 12) and* poly *is a polynomial function independent of $T(\mathcal{A})$.*

Proof. We prove the constrained $\mathcal{L}^{\mathrm{snd}}$-soundness of $\mathbf{PS}$ via a series of games. We give an overview of the games in Fig. 14.

$\mathbf{G} \rightsquigarrow \mathbf{G_0}$**:** Again we first remove the *lose*-functionality of the verification oracle for statements with $[\mathbf{c}] \in \mathcal{L}_{\mathrm{snd}} \backslash \mathcal{L}$. This can only raise the winning chances of the adversary.

$\mathbf{G_0} \rightsquigarrow \mathbf{G_1}$**:** The difference between $\mathbf{G}_0$ and $\mathbf{G}_1$ is that from game $\mathbf{G}_1$ on the oracle $\mathcal{O}_{\mathbf{ver}}$, on input $([\mathbf{c}], \Pi, \mathsf{pred})$, first checks if $[\mathbf{c}] \in \mathrm{span}([\mathbf{A}])$. If this is the case, $\mathcal{O}_{\mathbf{ver}}$ behaves as in game $\mathbf{G}_0$. Otherwise, it does not check if $[\pi^\star] = [\pi]$. As for the case $k = 1$, we show that $\varepsilon_1 \geq \varepsilon_0$. This transition follows from the fact that for $[\pi^\star] = \overline{[\mathbf{A}_0]} \cdot \mathbf{x} + \overline{[\mathbf{c}]} \cdot y$ we have $\underline{\mathbf{A}}_0 \overline{\mathbf{A}}_0^{-1} \left([\pi^\star] - \overline{[\mathbf{c}]} \cdot y \right) + \underline{[\mathbf{c}]} \cdot y = \underline{[\mathbf{A}_0]} \cdot \mathbf{x} + \underline{[\mathbf{c}]} \cdot y$, which agrees with the way keys that are computed in Game $\mathbf{G}_0$. Therefore - as the view on non-winning queries is unchanged - the winning chances of the adversary can only increase.

$\mathbf{G_1} \rightsquigarrow \mathbf{G_2}$**:** We set up the transition as in case $k = 1$ for both underlying proof systems $Pre\mathbf{PS}$. Let $([\mathbf{B}], [\mathbf{h}_1, \ldots, \mathbf{h}_{Q_{\mathbf{sim}}}])$ be a $Q_{\mathbf{sim}}$-fold $\mathcal{U}_{2k,k}$-MDDH challenge.

First $\mathcal{B}$ picks $\mathbf{A}, \mathbf{A}_0, \mathbf{A}_1$ as described in Section 4.1 and further for $j \in \{1, 2\}$ the adversary choses $\mathbf{K}'_{\mathbf{x},j} \leftarrow_R \mathbb{Z}_p^{(k+1)\times 2k}$ and $\mathbf{K}_{y,j} \leftarrow_R \mathbb{Z}_p^{2\times 2k}$.

The adversary $\mathcal{B}$ implicitely sets $\mathbf{K}_{\mathbf{x},j} = \mathbf{K}'_{\mathbf{x},j} + \mathbf{U}_j(\mathbf{A}^\perp)^\top$ for some $\mathbf{A}^\perp \in \mathsf{orth}(\mathbf{A})$ and $\mathbf{U}_j \in \mathbb{Z}_p^{(k+1)\times k}$, where the latter depends on the $\mathcal{U}_{2k,k}$-MDDH challenge (and cannot be computed by $\mathcal{B}$ explicitely). This will allow the adversy to embed the $\mathcal{U}_{2k,k}$-MDDH challenge into simulation

#	sim. $\mathbf{x}_j$ for $[\mathbf{c}] \in \mathcal{L}_{\mathrm{snd}} \backslash \mathcal{L}$	ver. $[\kappa_j]$ for $[\mathbf{c}] \notin \mathcal{L}$	game knows	remark
$\mathbf{G}_0$	$\mathbf{x}_j := \mathsf{h}_0\left(\mathbf{K}_{\mathbf{x},j}[\mathbf{c}]\right)$	$\underline{[\mathbf{A}_0]} \cdot \mathbf{x}_j + \underline{[\mathbf{c}]} \cdot y_j$		$\mathcal{L}_{\mathrm{snd}}$-soundn. game w/o *lose*
$\mathbf{G}_1$	$\mathbf{x}_j := \mathsf{h}_0\left(\mathbf{K}_{\mathbf{x},j}[\mathbf{c}]\right)$	$\underline{\mathbf{A}}_0\overline{\mathbf{A}}_0^{-1}\left([\pi_j^\star] - \overline{[\mathbf{c}]} \cdot y_j\right) + \underline{[\mathbf{c}]} \cdot y_j$	$\mathbf{A}, \mathbf{A}_0$	win. chances increase
$\mathbf{G}_2$	$\mathbf{u}_j \leftarrow_R \mathbb{Z}_p^{k+1}$, $\mathbf{x}_j := \mathsf{h}_0([\mathbf{u}_j])$	$\underline{\mathbf{A}}_0\overline{\mathbf{A}}_0^{-1}\left([\pi_j^\star] - \overline{[\mathbf{c}]} \cdot y_j\right) + \underline{[\mathbf{c}]} \cdot y_j$	$\mathbf{A}, \mathbf{A}_0$	$\mathcal{D}_{2k,k}$-MDDH
$\mathbf{G}_3$	$\mathbf{x}_j \leftarrow_R \mathbb{Z}_p^k$	$\underline{\mathbf{A}}_0\overline{\mathbf{A}}_0^{-1}\left([\pi_j^\star] - \overline{[\mathbf{c}]} \cdot y_j\right) + \underline{[\mathbf{c}]} \cdot y_j$	$\mathbf{A}, \mathbf{A}_0$	Lemma 1 (LOHL)

Fig. 14: Overview of the proof of $\mathcal{L}_{\mathrm{snd}}$-constrained soundness of **PS**. We give the changes of the underlying proof system *Pre***PS** for $j \in \{1, 2\}$. The first column shows how $\mathbf{x}$ is computed for queries to $\mathcal{O}_{\mathbf{sim}}$. The second column shows how the key $[\kappa]$ is computed by the verifier in queries to $\mathcal{O}_{\mathbf{ver}}$ when $[\mathbf{c}] \notin \mathcal{L}$. The third column "game knows" gives an overview of which non-public information need to be known by the game respective to $\mathbf{A}$, $\mathbf{A}_0$ and $\mathbf{A}_1$.

queries. Note that even without explicit knowledge of $\mathbf{K}_{\mathbf{x},j}$ the adversary can still compute the public parameters $[\mathbf{K}_{\mathbf{x},j}\mathbf{A}] = [\mathbf{K}'_{\mathbf{x},j}\mathbf{A}]$ and $[\mathbf{K}_{y,j}\mathbf{A}]$.

For $j \in \{1, 2\}$ the adversary $\mathcal{B}$ further chooses $\mathbf{W}_j \leftarrow \mathbb{Z}_p^{k \times k}$. This implicitly defines $[\mathbf{V}_j] := \mathbf{W}_j[\mathbf{B}\overline{\mathbf{B}}^{-1}]$. Note that $\mathcal{B}$ does not need to compute $[\mathbf{V}_j]$. Replacing $[(\mathbf{K}'_{\mathbf{x},j} + \mathbf{U}_j(\mathbf{A}^\perp)^\top)\mathbf{A}_0]$ by $[\mathbf{V}_j]$ is statistically indistinguishable for the adversary $\mathcal{A}$.

As in case $k = 1$ on the i-th query to $\mathcal{O}_{\mathbf{sim}}$, for all $i \in [Q_{\mathbf{dec}}]$, the adversary $\mathcal{B}$ defines $[\mathbf{c}] := \mathbf{A}_0\overline{[\mathbf{h}_i]}$ to be the challenge ciphertext and computes $\mathbf{x}_j := \mathsf{h}_0(\mathbf{W}_j[\mathbf{h}_i])$. In case of a real $\mathcal{U}_{2k,k}$-MDDH challenge, we have $\mathbf{h}_i = \mathbf{B}\mathbf{s}_i$ for $\mathbf{s}_i \leftarrow_R \mathbb{Z}_p^k$ and thus for $\mathbf{r}_i := \overline{\mathbf{B}}\mathbf{s}_i$ we have $[\mathbf{c}] = [\mathbf{A}_0\mathbf{r}_i]$ and $\mathbf{x}_j = \mathsf{h}_0(\mathbf{W}_j \cdot [\mathbf{B}\mathbf{s}_i]) = \mathsf{h}_0(\mathbf{W}_j \cdot [\mathbf{B}\overline{\mathbf{B}}^{-1}\mathbf{r}_i]) = \mathsf{h}_0([\mathbf{V}_j\mathbf{r}_i])$. By our previous considerations $[\mathbf{V}_j\mathbf{r}_i]$ is statistically close to $\mathbf{K}_{\mathbf{x},j}[\mathbf{c}]$ and thus adversary $\mathcal{B}$ simulates game $\mathbf{G}_1$. In case the adversary was given a random challenge, the vectors $\mathbf{h}_i$ are distributed uniformly over $\mathbb{Z}_p^{2k}$ and the adversary simulates game $\mathbf{G}_2$.

Finally, by Lemma 5 ($\mathcal{U}_{2k,k}$-MDDH $\Rightarrow$ $Q_{\mathbf{enc}}$-fold $\mathcal{U}_{2k,k}$-MDDH), and Lemma 3 ($\mathcal{D}_{2k,k}$-MDDH $\Rightarrow \mathcal{U}_{2k,k}$-MDDH), we obtain an adversary $\mathcal{B}$ such that $T(\mathcal{B}) \approx T(\mathcal{A}) + (Q_{\mathbf{ver}} + Q_{\mathbf{sim}}) \cdot \mathsf{poly}(\lambda)$ and

$$|\varepsilon_2 - \varepsilon_1| \leq \mathrm{Adv}^{\mathrm{mddh}}_{\mathbb{G},\mathcal{D}_{2k,k},\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}.$$

$\mathbf{G_2} \rightsquigarrow \mathbf{G_3}$**:** As in case $k = 1$ we have

$$|\varepsilon_2 - \varepsilon_3| \leq 2 \cdot Q_{\mathbf{sim}}/p.$$

Game $\mathbf{G_3}$: Again we argue that by queries to $\mathcal{O}_{\mathbf{sim}}$ and the first i queries to $\mathcal{O}_{\mathbf{ver}}$ the adversary can only learn valid tuples $([\mathbf{c}], [\pi], [\kappa])$ with $[\mathbf{c}] \in \mathcal{L}_{\mathrm{snd}}$ which do not reveal more about $\mathbf{K}_{y,1}, \mathbf{K}_{y,2}$ than the public key leaks informationtheoretically.

For any $[\mathbf{c}] \notin \mathcal{L}_{\mathrm{snd}}$, $y_1 = \mathsf{h}_1[\mathbf{K}_{y,1}\mathbf{c}]$ and $y_2 = \mathsf{h}_1[\mathbf{K}_{y,2}\mathbf{c}]$ computed by $\mathcal{O}_{\mathbf{ver}}$ are thus distributed statistically close to uniform from the adversary's point of view. Thus the keys $[\kappa_1]$ and $[\kappa_2]$ carry each $\log p$ bits of entropy and the universality of h_2 allows us to employ the Left Over Hash Lemma

$\widetilde{\mathbf{PGen}}(1^\lambda)$:

$(ppk_1, psk_1) \leftarrow Pre\mathbf{PGen}(1^\lambda)$
$(ppk_2, psk_2) \leftarrow Pre\mathbf{PGen}(1^\lambda)$
return
  $ppk := (ppk_1, ppk_2)$
  $psk := (psk_1, psk_2)$

$\widetilde{\mathbf{PVer}}(ppk, psk, [\mathbf{c}], [\pi^\star])$:

$[\pi_1, \kappa_1] := \widetilde{Pre\mathbf{PSim}}(ppk_1, psk_1, [\mathbf{c}])$
$[\pi_2, \kappa_2] := \widetilde{Pre\mathbf{PSim}}(ppk_2, psk_2, [\mathbf{c}])$
if $[\pi_1, \pi_2] = [\pi^\star]$
  return $(1, [\mathsf{h}_2([\kappa_1, \kappa_2])])$
else return $(0, \perp)$

$\widetilde{\mathbf{PPrv}}(ppk, [\mathbf{c}], \mathbf{r})$:

$[\pi_1, \kappa_1] := \widetilde{Pre\mathbf{PPrv}}(ppk_1, [\mathbf{c}], \mathbf{r})$
$[\pi_2, \kappa_2] := \widetilde{Pre\mathbf{PPrv}}(ppk_2, [\mathbf{c}], \mathbf{r})$
return
  $[\pi] := [\pi_1, \pi_2]$
  $[\kappa] := [\mathsf{h}_2([\kappa_1, \kappa_2])]$

$\widetilde{\mathbf{PSim}}(ppk, psk, [\mathbf{c}])$:

$[\pi_1, \kappa_1] := \widetilde{Pre\mathbf{PSim}}(ppk_1, psk_1, [\mathbf{c}])$
$[\pi_2, \kappa_2] := \widetilde{Pre\mathbf{PSim}}(ppk_2, psk_2, [\mathbf{c}])$
return
  $[\pi] := [\pi_1, \pi_2]$
  $[\kappa] := [\mathsf{h}_2([\kappa_1, \kappa_2])]$

Fig. 15: $\widetilde{\mathcal{L}_{\text{snd}}}$-qualified Proof System $\widetilde{\mathbf{PS}}$ for $\mathcal{L}$ in case $k > 1$, where $\widetilde{Pre\mathbf{PS}}$ is defined as in Fig. 4.

to obtain that $[\kappa] := \mathsf{h}_2([\kappa_1, \kappa_2])$ is statistically close to uniform over $\mathbb{G}$. Now a hybrid argument as for $k = 1$ yields

$$\varepsilon_3 \leq Q_{\mathbf{ver}} \cdot \text{uncert}^{\text{snd}}_{\mathcal{A}}(\lambda) + Q_{\mathbf{ver}} \cdot 2^{-\Omega(\lambda)}.$$

□

We omit the proof of the following two lemmas, as the proof techniques are similar to the ones presented in Lemma 7, Lemma 8 and Theorem 2.

Lemma 11 ($\mathcal{L}_{\text{snd}}$-indistinguishability). *For $k > 1$ proof systems $\mathbf{PS}$ and $\widetilde{\mathbf{PS}}$ described in Fig. 5 and Fig. 15, resp., are $\mathcal{L}_{\text{snd}}$-indistinguishable. That is, for every (unbounded) adversary $\mathcal{A}$ we have $\text{Adv}^{\text{PS-ind}}_{\mathcal{L}_{\text{snd}},\mathbf{PS},\widetilde{\mathbf{PS}},\mathcal{A}}(\lambda) = 2^{-\Omega(\lambda)}$.*

Lemma 12 (Constrained $\widetilde{\mathcal{L}_{\text{snd}}}$-soundness of $\widetilde{\mathbf{PS}}$ for $k > 1$). *If the $\mathcal{D}_{2k,k}$-MDDH assumption holds in $\mathbb{G}$ and h_0, h_1 and h_2 are universal hash functions, then the proof system described in Fig. 15 complies with constrained $\widetilde{\mathcal{L}_{\text{snd}}}$-soundness. Namely, for any adversary $\mathcal{A}$ against $\widetilde{\mathcal{L}_{\text{snd}}}$-soundness, there exists an adversary $\mathcal{B}$ such that $T(\mathcal{B}) \approx T(\mathcal{A}) + (Q_{\mathbf{dec}} + Q_{\mathbf{ver}}) \cdot \mathsf{poly}(\lambda)$ and*

$$\text{Adv}^{\text{csnd}}_{\widetilde{\mathcal{L}_{\text{snd}}},\widetilde{\mathbf{PS}},\mathcal{A}}(\lambda) \leq 2 \cdot \text{Adv}^{\text{mddh}}_{\mathbb{G},\mathcal{B},\mathcal{D}_{2k,k}}(\lambda) + 2 \cdot Q_{\mathbf{ver}} \cdot \text{uncert}_{\mathcal{A}}(\lambda) + (Q_{\mathbf{sim}} + Q_{\mathbf{ver}}) \cdot 2^{-\Omega(\lambda)},$$

where $Q_{\mathbf{ver}}$, $Q_{\mathbf{dec}}$ are the number of calls to $\mathcal{O}_{\mathbf{ver}}$ and $\mathcal{O}_{\mathbf{dec}}$ respectively, $\text{uncert}^{\text{snd}}_{\mathcal{A}}(\lambda)$ describes the uncertainty of the predicates provided by $\mathcal{A}$ and poly is a polynomial function, independent of $T(\mathcal{A})$.

B Security in the Multi-User Setting

For the sake of better readability we merely considered security in the single-user setting so far. In this section, we want to give an idea on how to carry over our results to the multi-user setting. In the following we give the alterations in the IND-CCCA security definition of a key encapsulation mechanism (Definition 10) for the multi-user setting.

$Exp^{\mathrm{mu-ccca}}_{\mathbf{KEM},\mathcal{A}}(\lambda)$:	$\mathcal{O}_{\mathbf{enc}}(j)$:	$\mathcal{O}_{\mathbf{dec}}(j, \mathsf{pred}_i, C_i)$:
$(pk_j, sk_j)_j \leftarrow_R \mathbf{KGen}(1^\lambda)$	$K_0 \leftarrow_R \mathcal{K}(\lambda)$	$K_i := \mathbf{KDec}(sk_j, C_i)$
$b \leftarrow_R \{0,1\}$	$(C, K_1) \leftarrow_R \mathbf{KEnc}(pk_j)$	$C_i \notin \mathcal{C}^j_{\mathbf{enc}}$ **and**
$\mathcal{C}_{\mathbf{enc}} := \emptyset$	$\mathcal{C}^j_{\mathbf{enc}} := \mathcal{C}^j_{\mathbf{enc}} \cup \{C\}$	**if** $\mathsf{pred}_i(K_i) = 1$
$b' \leftarrow_R \mathcal{A}^{\mathcal{O}_{\mathbf{enc}},\mathcal{O}_{\mathbf{dec}}}((pk_j)_j)$	**return** (C, K_b)	**return** K_i
if $b = b'$ **return** 1		**else return** $\perp$
else return 0		

We start the security analysis in the multi-user setting by adapting the security of the proof system **PS** presented in Figure 5. We omit transferring the definitions of qualified proof system to the multi-user case, as it is straightforward. The only point in the proof of security and extensibility of **PS** that is not statistical and hence needs to be adapted is the transition from game $\mathbf{G}_1$ to $\mathbf{G}_2$ in the proof of Theorem 2. Here we use the $\mathcal{U}_{2k,k}$-MDDH assumption to tightly switch $([\mathbf{B}], [\mathbf{Bh}])$ to $([\mathbf{B}], [\mathbf{u}])$ for a uniformly random matrix $[\mathbf{B}]$. Reusing a technique presented in the proof of Theorem 3, we can rerandomize given $\mathcal{U}_{2k,k}$-MDDH tuples $([\mathbf{B}], [\mathbf{z}])$ to $([\mathbf{B}_j], [\mathbf{z}_j])$ and thereby tightly perform this step for all users simultaneously.

For the adaptation of Theorem 3 and Lemma 9 either the same technique can be employed (e.g. in the transition $\mathbf{G}_4 \rightsquigarrow \mathbf{G}_5$) or MDDH is only applied on public parameters, which are the same for all users. The remaining transitions are statistical or rely on properties of the proof system **PS** and thus need not be studied anew.