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.

*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.

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).

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

[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].

[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.

Reference	$\lvert pk\rvert$	$\lvert c\rvert - \lvert m\rvert$	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.

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.

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]

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

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

[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 $[\mathbf{x}, \mathbf{y}] := ([\mathbf{x}], [\mathbf{y}])$.

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 $\mathbf{c} \notin \operatorname{span}(\mathbf{A})$, 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 $\mathbf{c} \in \operatorname{span}(\mathbf{A})$ only reveals information about $\mathbf{k}_0, \mathbf{k}_1$ that is already contained in pk. On the other hand, any decryption query with $\mathbf{c} \notin \operatorname{span}(\mathbf{A})$ 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 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 ci-

[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.

phertexts with $\mathbf{c} \in \operatorname{span}(\mathbf{A})$ (since $\mathbf{k}_0^\top \mathbf{A}\mathbf{r} = \mathbf{k}_0'^\top \mathbf{A}\mathbf{r}$), and randomize all challenge ciphertexts (since their $\mathbf{c}$ satisfies $\mathbf{c} \notin \operatorname{span}(\mathbf{A})$ 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 $\mathbf{c} \notin \operatorname{span}(\mathbf{A})$. 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:

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

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

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

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 $\mathbf{c} \notin \operatorname{span}(\mathbf{A})$) 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 $\mathbf{c} \notin \operatorname{span}(\mathbf{A})$), 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 $\operatorname{span}(\mathbf{A}_{\tau_{i+1}})$, 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 $\mathbf{c} \in \operatorname{span}(\mathbf{A}_{\tau_{i+1}})$, 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

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

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*

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

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*

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

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*

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

where $\varDelta$ 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 $\mathbb{G}$:

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

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 $([\mathbf{A}], [\mathbf{Aw}])$ 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 $\mathcal{A}$,*

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

$$\leq \mathsf{negl}(\lambda),$$

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

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 $([\mathbf{A}], [\mathbf{AW}])$ 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*

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

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 $\ell \geq k$, 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 ℓ, 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*

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

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 $\mathcal{A}$,*

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

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:*

```
$\underline{\mathrm{Exp}^{\mathrm{cca}}_{\mathbf{PKE},\mathcal{A}}(\lambda)}$:
$(pk, sk) \leftarrow_R \mathbf{Gen}(1^\lambda)$
$b \leftarrow_R \{0,1\}$
$\mathcal{C}_{\mathbf{enc}} := \emptyset$
$b' \leftarrow_R \mathcal{A}^{\mathcal{O}_{\mathbf{enc}}(\cdot,\cdot),\mathcal{O}_{\mathbf{dec}}(\cdot)}(pk)$
if $b = b'$ return $1$
else return $0$
```

```
$\underline{\mathcal{O}_{\mathbf{enc}}(M_0, M_1)}$:
if $|M_0| = |M_1|$
    $C \leftarrow_R \mathbf{Enc}(pk, M_b)$
    $\mathcal{C}_{\mathbf{enc}} := \mathcal{C}_{\mathbf{enc}} \cup \{C\}$
    return $C$
```

```
$\underline{\mathcal{O}_{\mathbf{dec}}(C)}$:
if $C \notin \mathcal{C}_{\mathbf{enc}}$
    $M := \mathbf{Dec}(pk, sk, C)$
    return $M$
else return $\perp$
```

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.*

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

[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].

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:*

```
$\mathrm{Exp}^{\mathrm{ccca}}_{\mathbf{KEM},\mathcal{A}}(\lambda)$:
$(pk, sk) \leftarrow_R \mathbf{KGen}(1^\lambda)$
$b \leftarrow_R \{0,1\}$
$\mathcal{C}_{\mathbf{enc}} := \emptyset$
$b' \leftarrow_R \mathcal{A}^{\mathcal{O}_{\mathbf{enc}}, \mathcal{O}_{\mathbf{dec}}(\cdot,\cdot)}(pk)$
if $b = b'$ return $1$
else return $0$
```

```
$\mathcal{O}_{\mathbf{enc}}$:
$K_0 \leftarrow_R \mathcal{K}(\lambda)$
$(C, K_1) \leftarrow_R \mathbf{KEnc}(pk)$
$\mathcal{C}_{\mathbf{enc}} := \mathcal{C}_{\mathbf{enc}} \cup \{C\}$
return $(C, K_b)$
```

```
$\mathcal{O}_{\mathbf{dec}}(\mathsf{pred}_i, C_i)$:
$K_i := \mathbf{KDec}(pk, sk, C_i)$
if $C_i \notin \mathcal{C}_{\mathbf{enc}}$ and
  if $\mathsf{pred}_i(K_i) = 1$
    return $K_i$
else return $\perp$
```

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 $\text{uncert}_{\mathcal{A}}(\lambda)$ 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 $\text{uncert}_{\mathcal{A}}(\lambda)$ 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:*

PGen(1^λ)*: generates a public key ppk and a secret key psk.*
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.*
PVer(ppk, psk, x, Π)*: 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 $\perp$.*
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 $\mathcal{L}^{\mathrm{snd}} \setminus \mathcal{L}$ 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 $\mathcal{L}^{\mathrm{snd}}$. The adversary loses immediately if it provides a valid verification query in $\mathcal{L}^{\mathrm{snd}} \setminus \mathcal{L}$. 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 $\mathcal{L}^{\mathrm{snd}} \setminus \mathcal{L}$ 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* $\mathcal{L}^{\mathrm{snd}} = \{\mathcal{L}^{\mathrm{snd}}_{pars}\}$ *be a family of languages, such that* $\mathcal{L}_{pars} \subseteq \mathcal{L}^{\mathrm{snd}}_{pars}$. *We say that* $\mathbf{PS}$ *is* $\mathcal{L}^{\mathrm{snd}}$-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* $(\Pi, K) \coloneqq \mathbf{PPrv}(ppk, x, w)$.

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

```
$\mathrm{Exp}^{\mathrm{csnd}}_{\mathbf{PS},\mathcal{A}}(\lambda)$:
$(ppk, psk) \leftarrow_R \mathbf{PGen}(1^\lambda)$
$\mathcal{A}^{\mathcal{O}_{\mathbf{sim}},\mathcal{O}_{\mathbf{ver}}(\cdot,\cdot,\cdot)}(1^\lambda, ppk)$
if $\mathcal{O}_{\mathbf{ver}}$ returned lose
    return $0$
if $\mathcal{O}_{\mathbf{ver}}$ returned win
    return $1$
return $0$
```

```
$\mathcal{O}_{\mathbf{sim}}$:
$x \leftarrow_R \mathcal{L}^{\mathrm{snd}} \backslash \mathcal{L}$
$(\Pi, K) \leftarrow \mathbf{PSim}(psk, x)$
return $(x, \Pi, K)$
```

```
$\mathcal{O}_{\mathbf{ver}}(x, \Pi, \mathsf{pred})$:
$(v, K) := \mathbf{PVer}(psk, x, \Pi)$
if $v = 1$ and $\mathsf{pred}(K) = 1$
    if $x \in \mathcal{L}$
        return $K$
    else if $x \in \mathcal{L}^{\mathrm{snd}}$
        return lose and abort
    else return win and abort
else return $\bot$
```

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, Π, pred) of $\mathcal{A}$ with $x \notin \mathcal{L}^{\mathrm{snd}}$ and the following conditions hold:*

- *The predicate corresponding to the i-th query is of the form $\mathsf{pred}_i \colon \mathcal{K} \cup \{\bot\} \to \{0,1\}$ with $\mathsf{pred}_i(\bot) = 0$ 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*

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

  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 $\mathrm{Adv}^{\mathrm{csnd}}_{\mathcal{L}^{\mathrm{snd}},\mathbf{PS},\mathcal{A}}(\lambda) := \Pr[\mathrm{Exp}^{\mathrm{csnd}}_{\mathbf{PS},\mathcal{A}}(\lambda) = 1]$, where the probability is taken over the random coins of $\mathcal{A}$ and $\mathrm{Exp}^{\mathrm{csnd}}_{\mathbf{PS},\mathcal{A}}$. Then we say* constrained $\mathcal{L}^{\mathrm{snd}}$-soundness *holds for $\mathbf{PS}$, if for every PPT adversary $\mathcal{A}$, $\mathrm{Adv}^{\mathrm{csnd}}_{\mathcal{L}^{\mathrm{snd}},\mathbf{PS},\mathcal{A}}(\lambda) = \mathsf{negl}(\lambda)$.*

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 ($\mathcal{L}^{\mathrm{snd}}$-indistinguishability of two proof systems). *Let $\mathcal{L} \subseteq \mathcal{L}^{\mathrm{snd}}$ 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):*

```
$\mathrm{Exp}^{\mathrm{PS-ind}}_{\mathcal{L}^{\mathrm{snd}},\mathbf{PS}_0,\mathbf{PS}_1,\mathcal{A}}(\lambda)$:
$b \leftarrow_R \{0,1\}$
$(ppk, psk) \leftarrow \mathbf{PGen}_b(1^\lambda)$
$b' \leftarrow \mathcal{A}^{\mathcal{O}^b_{\mathbf{sim}},\mathcal{O}^b_{\mathbf{ver}}(\cdot,\cdot)}(ppk)$
if $b = b'$ return $1$
else return $0$
```

```
$\mathcal{O}^b_{\mathbf{sim}}$:
$x \leftarrow_R \mathcal{L}^{\mathrm{snd}} \backslash \mathcal{L}$
$(\Pi, K) \leftarrow \mathbf{PSim}_b(psk, x)$
return $(x, \Pi, K)$
```

```
$\mathcal{O}^b_{\mathbf{ver}}(x, \Pi, \mathsf{pred})$:
$(v, K) := \mathbf{PVer}_b(psk, x, \Pi)$
if $v = 1$ and $\mathsf{pred}(K) = 1$
and $x \in \mathcal{L}^{\mathrm{snd}}$
    return $K$
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

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

We assume from now that all algorithms have access to $\mathit{pars}_{\mathbf{PS}}$ 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 A Construction based on MDDH

The goal of this section 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 $\mathit{Pre}\mathbf{PS} := (\mathit{Pre}\mathbf{PGen}, \mathit{Pre}\mathbf{PPrv}, \mathit{Pre}\mathbf{PVer}, \mathit{Pre}\mathbf{PSim})$ for $\mathcal{L}$ in Fig. 2.

In case $k = 1$ this is sufficient, namely setting $\mathbf{PGen} := \mathit{Pre}\mathbf{PGen}$, $\mathbf{PPrv} := \mathit{Pre}\mathbf{PPrv}$, $\mathbf{PVer} := \mathit{Pre}\mathbf{PVer}$ and $\mathbf{PSim} := \mathit{Pre}\mathbf{PSim}$, 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 the full version.

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).

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

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

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

$\mathit{Pre}\mathbf{PSim}(\mathit{ppk}, \mathit{psk}, [\mathbf{c}])$:
$\mathbf{x} := \mathsf{h}_0(\mathbf{K_x}[\mathbf{c}]) \in \mathbb{Z}_p^k$
$y := \mathsf{h}_1(\mathbf{K}_y[\mathbf{c}]) \in \mathbb{Z}_p$
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 $\mathit{Pre}\mathbf{PS}$ for $\mathcal{L}$. For $k = 1$ the proof system $\mathbf{PS} := \mathit{Pre}\mathbf{PS}$ is $\mathcal{L}_{\text{snd}}$-qualified based on DDH.

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 $\mathbf{PS} := \mathit{Pre}\mathbf{PS}$ 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.

In the full version we prove that $\mathbf{PS}$ is $\widetilde{\mathcal{L}^{\text{snd}}}$-extensible. □

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 $\mathcal{L}_{\mathrm{snd}}$-soundness of $\mathbf{PS}$ to prove IND-CCCA security of our KEM), the term $\mathrm{uncert}_{\mathcal{A}}^{\mathrm{snd}}(\lambda)$ will be statistically small, so we can afford to get a security loss of $Q_{\mathbf{ver}} \cdot \mathrm{uncert}_{\mathcal{A}}^{\mathrm{snd}}(\lambda)$ without compromising tightness.

Proof. We prove $\mathcal{L}_{\mathrm{snd}}$*-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 $[\mathbf{c}] \in \mathcal{L}_{\mathrm{snd}} \setminus \mathcal{L}$, 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 $[\mathbf{c}] \in \mathcal{L}_{\mathrm{snd}}$, 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 $[\mathbf{c}] \in \mathcal{L}_{\mathrm{snd}}$, an adversary cannot do better than guessing y to produce a valid key for a statement outside $\mathcal{L}_{\mathrm{snd}}$, and therefore, only has negligible winning chances.

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

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

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

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

$\mathbf{G} \rightsquigarrow \mathbf{G_0}$**:** From game $\mathbf{G}_0$ on, on a valid verification query $([\mathbf{c}], \mathit{\Pi}, \mathsf{pred})$ 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 $([\mathbf{c}], \mathit{\Pi}, \mathsf{pred})$, first

#	**sim.** x **for** $[\mathbf{c}] \in \mathcal{L}_{\mathrm{snd}} \backslash \mathcal{L}$	**ver.** $[\kappa]$ **for** $[\mathbf{c}] \notin \mathcal{L}$	**game knows**	**remark**
$\mathbf{G}_0$	$x := \mathsf{h}_0\left(\mathbf{K}_x[\mathbf{c}]\right)$	$\underline{[\mathbf{A}_0]} \cdot \mathbf{x} + \underline{[\mathbf{c}]} \cdot y$		$\mathcal{L}_{\mathrm{snd}}$-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. 3: Overview of the proof of $\mathcal{L}_{\mathrm{snd}}$-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 $[\mathbf{c}] \notin \mathcal{L}$.

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 $[\pi^\star] = [\pi]$ 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 $\mathcal{L}_{\mathrm{snd}} \backslash \mathcal{L}$, and thus games $\mathbf{G}_0$ and $\mathbf{G}_1$ only differ when $\mathcal{O}_{\mathbf{ver}}$ is queried on statements with $[\mathbf{c}] \notin \mathcal{L}_{\mathrm{snd}}$. Therefore it remains to show that for any query $([\mathbf{c}], [\pi^\star], \mathsf{pred})$ to $\mathcal{O}_{\mathbf{ver}}$ with $[\mathbf{c}] \notin \mathcal{L}_{\mathrm{snd}}$, we have that if the query is winning in $\mathbf{G}_0$, then it is also winning in $\mathbf{G}_1$. Suppose $([\mathbf{c}], [\pi^\star], \mathrm{pred})$ 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 $x \in \mathcal{L}^{\mathrm{snd}} \backslash \mathcal{L}$, 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 $x \in \mathcal{L}^{\mathrm{snd}} \backslash \mathcal{L}$ 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:

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

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 the full version). 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. 4.

Efficiency. When using our qualified proof system from Section 4 (respectively for $k > 1$ from the full version) 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,

```
$\mathbf{KGen}(1^\lambda)$:
$(ppk, psk) \leftarrow_R \mathbf{PGen}(1^\lambda)$
$\mathbf{k}_0, \mathbf{k}_1 \leftarrow_R \mathbb{Z}_p^{2k}$
return
    $pk := (ppk, [\mathbf{k}_0^\top \mathbf{A}], [\mathbf{k}_1^\top \mathbf{A}])$
    $sk := (psk, \mathbf{k}_0, \mathbf{k}_1)$
```

```
$\mathbf{KEnc}(pk)$:
$\mathbf{r} \leftarrow_R \mathbb{Z}_p^k$
$[\mathbf{c}] := [\mathbf{A}]\mathbf{r}$
$(\Pi, [\kappa]) := \mathbf{PPrv}(ppk, [\mathbf{c}], \mathbf{r})$
$\tau := \mathsf{H}([\mathbf{c}])$
return
    $C := ([\mathbf{c}], \Pi)$
    $K := ([\mathbf{k}_0^\top \mathbf{A}] + \tau[\mathbf{k}_1^\top \mathbf{A}])\mathbf{r} + [\kappa]$

$\mathbf{KDec}(pk, sk, C)$ :
parse $C := ([\mathbf{c}], \Pi)$
$(b, [\kappa]) := \mathbf{PVer}(psk, [\mathbf{c}], \Pi)$
if $b = 0$ return $\perp$
$\tau := \mathsf{H}([\mathbf{c}])$
return $K := (\mathbf{k}_0 + \tau\mathbf{k}_1)^\top [\mathbf{c}] + [\kappa]$
```

Fig. 4: Construction of the KEM

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 2 (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* $\mathbf{KEM}$ *described in Fig. 4 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. 5.

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 can be found in the full version.

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

$\mathbf{G_0} \rightsquigarrow \mathbf{G_1}$: 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 $\mathcal{A}$

#	ch. $\mathbf{c}$	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, $[\mathbf{c}] \in \text{span}([\mathbf{A}])$	see full version
$\mathbf{G}_5$	$\mathbf{A}_0$	rand	τ fresh, $[\mathbf{c}] \in \text{span}([\mathbf{A}])$	$\mathcal{D}_{2k,k}$-MDDH

Fig. 5: 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 $\perp$.

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 $\text{span}([\mathbf{A}])$, 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

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

$\mathbf{G_1} \rightsquigarrow \mathbf{G_2}$: 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 $\mathbf{PS}$, 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 $\mathrm{span}([\mathbf{A}])$ 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 $([\mathbf{A}], [\mathbf{v}_1|\ldots|\mathbf{v}_{Q_{\mathbf{enc}}}])$ with $[\mathbf{A}] \in \mathbb{G}^{2k \times k}$ and $[\mathbf{V}] := [\mathbf{v}_1|\ldots|\mathbf{v}_{Q_{\mathbf{enc}}}] \in \mathbb{G}^{2k \times Q_{\mathbf{enc}}}$ 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 the full version.

$\mathbf{G_4} \rightsquigarrow \mathbf{G_5}$**:** 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 $([\mathbf{B}], [\mathbf{h}_1|\ldots|\mathbf{h}_{Q_{\mathbf{enc}}}])$ with $\mathbf{B} \leftarrow_R \mathcal{U}_k$ (that is $\mathbf{B} \in \mathbb{Z}_p^{(k+1)\times k}$) and $\mathbf{h}_1, \ldots, \mathbf{h}_{Q_{\mathbf{enc}}} \in \mathbb{Z}_p^{k+1}$, $\mathcal{B}$ 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

$$\varepsilon_5 = 0.$$

□

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 Secu-

rity". 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.