A Twist on the Naor-Yung Paradigm and Its Application to Efficient CCA-Secure Encryption from Hard Search Problems

Ronald Cramer[*], Dennis Hofheinz[**], and Eike Kiltz[***]

Abstract. The Naor-Yung (NY) paradigm shows how to build a chosen-ciphertext secure encryption scheme from three conceptual ingredients:

- a weakly (i.e., IND-CPA) secure encryption scheme,
- a "replication strategy" that specifies how to use the weakly secure encryption scheme; concretely, a NY-encryption contains several weak encryptions of the same plaintext,
- a non-interactive zero-knowledge (NIZK) proof system to show that a given ciphertext is consistent, i.e., contains weak encryptions of the *same* plaintext.

The NY paradigm served both as a breakthrough proof-of-concept, and as an inspiration to subsequent constructions. However, the NY construction leads to impractical encryption schemes, due to the usually prohibitively expensive NIZK proof.

In this contribution, we give a variant of the NY paradigm that leads to practical, fully IND-CCA secure encryption schemes whose security can be based on a generic class of algebraic complexity assumptions. Our approach refines NY's approach as follows:

- Our sole computational assumption is that of a Diffie-Hellman (DH) type two-move key exchange protocol, interpreted as a weakly secure key encapsulation mechanism (KEM).
- Our "replication strategy" is as follows. Key generation consists of replicating the KEM several times, but *only the first pass*. Encryption then consists of performing the second pass with respect to all of these, *but with the same random coins* in each instance.
- For proving consistency of a given ciphertext, we employ a practical universal hash proof system, case-tailored to our KEM and replication strategy.

We instantiate our paradigm both from *computational* Diffie-Hellman (CDH) and from RSA type assumptions. This way, practical IND-CCA secure encryption schemes based on *search* problems can be built and explained in a generic, NY-like fashion.

We would like to stress that while we generalize universal hash proof systems *as a proof system*, we do *not* follow or generalize the approach of Cramer and Shoup to build IND-CCA secure encryption. Their approach uses specific hash proof systems that feature, on top of a NIZK property, a computational indistinguishability property. Hence they necessarily build upon *decisional* assumptions, whereas we show how to implement our approach with *search* assumptions. Our approach uses hash proof systems in the NY way, namely solely as a device to prove consistency. In our case, secrecy is provided by the "weak encryption" component, which allows us to embed search problems.

Keywords: public-key encryption, chosen-ciphertext security.

[*] CWI, Amsterdam, and Leiden University. cramer@cwi.nl.

[**] Karlsruhe Institute of Technology. Dennis.Hofheinz@kit.edu. Work performed while the author was with CWI. Supported by NWO.

[***] CWI, Amsterdam. kiltz@cwi.nl. Supported by the research program Sentinels.

1 Introduction

One of the main fields of interest in cryptography is the design and the analysis of the security of encryption schemes in the public-key setting (PKE schemes). The notion of security against chosen-ciphertext attack (IND-CCA security) is due to Rackoff and Simon [23] and is now widely accepted as the standard security notion for public-key encryption schemes. In contrast to security against passive adversaries (security against chosen-plaintext attacks aka semantic security), in a chosen-ciphertext attack the adversary plays an active role by obtaining the decryptions of ciphertexts (or even arbitrary bit-strings) of his choosing. The practical significance of such attacks was demonstrated by Bleichenbacher [1] by means of an IND-CCA attack against schemes following the encryption standard PKCS #1.

The Naor-Yung paradigm. Historically, the first scheme that was provably secure against a weaker variant of IND-CCA attacks (namely, "lunch-time attacks") is due to Naor and Yung (NY) [21]. Dolev, Dwork, and Naor [9] later showed how to modify the paradigm of NY to achieve full IND-CCA security. We briefly outline the general idea in the following. To start, let's assume a weakly (i.e., IND-CPA) secure encryption scheme. (IND-CPA secure encryption schemes are a well-understood primitive, and can be constructed from various search or decisional computational problems, see, e.g., [11].) Now a ciphertext contains two weakly secure encryptions of *the same message* (under different public keys of the weakly secure scheme), along with a non-interactive zero-knowledge (NIZK) proof that indeed the same messages were encrypted. During the security proof, a simulator will know precisely one of the two secret keys for the weakly secure encryption schemes. (Note that in order to implement the decryption oracle, the simulator only needs to decrypt one ciphertext component and rely on the soundness of the NIZK proof.) Hence, we can carve out three conceptual ingredients for the NY paradigm:

- a weakly (i.e., IND-CPA) secure encryption scheme,
- a "replication strategy" that specifies how to use the weakly secure encryption scheme, and
- a NIZK proof system to show that a given ciphertext is consistent.

Of course, these ingredients are not independent (e.g., the statement to be proven by the NIZK proof depends both on the weakly secure encryption scheme and on the replication strategy). The system of Dolev, Dwork and Naor uses a similar overall approach, the main difference the used replication strategy. (Additionally, they also require a more special type of "non-malleable" NIZK proof systems.)

Hash proof systems. The first *practical* schemes provably IND-CCA secure under standard cryptographic hardness assumptions were due to Cramer and Shoup [8,8]. They later generalized their initial scheme to the paradigm of "hash proof systems" (HPSs) [7], thereby yielding new practical schemes from a number of alternative intractability assumptions. The approach of Cramer and Shoup is inspired by the NY paradigm. And indeed, a HPS, as used by Cramer and Shoup, combines the encryption and the proof part of the NY paradigm in one primitive. However, even though the concept of HPSs is generic, its use in [7] to build encryption schemes inherently relies on *decisional assumptions*, such as the assumed hardness of deciding if a given integer has a square root modulo a composite number with unknown factorization, or if deciding if a given tuple is a Diffie-Hellman tuple or not (DDH assumption).

Particular instances of the HPS-based schemes could later be optimized, leading to a number of even more efficient schemes (e.g., [10,19,14,18]). However, all of these schemes are based on decisional assumptions (mostly the DDH assumption).

Lossy trapdoor functions and the approach of Rosen and Segev. An alternative generic framework of constructing IND-CCA secure encryption schemes is given by the recent concept of lossy trapdoor functions [22] that led to the first construction based on a (decisional) assumption related to finding shortest vectors on lattices. However, also lossy trapdoor functions inherently rely on decisional assumptions rather than computational assumptions.[1]

Recently, Rosen and Segev [24] proposed a refinement of the NY approach that does *not* require an explicit NIZK part. (In fact, consistency of a ciphertext can be checked deterministically be the decryptor, since they employ weak encryption schemes that are actually *functions*.) Unfortunately, their approach requires a nonstandard computational assumption, namely one-way security of several independent functions under *correlated inputs*. They show that security under correlated inputs is implied by lossy trapdoor functions; however, they do not show security under a standard *search* assumption.

IND-CCA security from identity-based encryption. Boneh et al [3] describe a completely generic transformation of a selective-ID secure identitity-based encryption scheme into an IND-CCA secure PKE scheme. Their assumption (a selective-ID secure IBE scheme) cannot be directly counted as a search or decisional assumption. However, as it is based upon indistinguishability of adversarial views, it is arguably closer to a decisional assumption.

Decisional vs. search assumptions. We conclude that all known *generic paradigms* of constructing practical IND-CCA secure encryption seem to rely on *decisional* assumptions, as opposed to *search* assumptions. No generic paradigm is known under which practical IND-CCA secure schemes based on, say, the CDH problem could be constructed or explained.

In most known cases related to cryptography, decisional assumptions form a much stronger class of assumptions than the corresponding search assumptions. For example, deciding if a given integer has a modular square root or not may be much easier than actually computing a square root (or, equivalently, factoring the modulus). Only recently were practical schemes proposed whose IND-CCA security does not rely on decisional assumptions (e.g., [3,5,13,16]). In particular, the first practical encryption scheme IND-CCA secure under the *Computational*

[1] Unless, of course, the decisional assumption can be proved equivalent to a computational assumption, as it is the case with cryptosystems based on the problem of "learning with error" [22].

Diffie-Hellman (CDH) assumption was proposed by Cash, Kiltz, and Shoup [5] in 2008, and improved by Hanaoka and Kurosawa [13] later that year. In 2009, Hofheinz and Kiltz proposed a very efficient IND-CCA secure encryption scheme under the factoring assumption [16].

However, there seems to be no overarching concept that explains these schemes. Each of these schemes relies on different techniques to achieve security, and in particular to conduct a reduction in the security proof.

Our contribution. In this work, we refine the abstract NY paradigm in a way that allows to construct and explain *practical* IND-CCA secure encryption schemes whose security is based on general widely believed *search* assumptions. Concretely, we modify the NY paradigm as follows:

- Our sole computational assumption is that of a Diffie-Hellman type two-move key exchange protocol. This assumption is implied, e.g., under the CDH assumption in cyclic groups, or under the RSA assumptions. We interpret the key exchange protocol as a weakly secure key encapsulation mechanism. (That is, the first KE message is the KEM public key, and the second KE message is the KEM ciphertext.)
- Our "replication strategy" uses the above KEM several times, but with the same *encryption random coins* (and not with the same key or plaintext as in the original NY paradigm). Our replication strategy comes with a special simulation setup, such that the simulator in the security proof can decrypt *all* consistent ciphertexts, except for *one* predetermined target/challenge ciphertext.
- For proving consistency of a given ciphertext, we employ a generic but practical universal hash proof system. This forms a special type of designated-verifier NIZK proof system, case-tailored to our KEM and replication strategy. We stress that we do *not* use the HPS in the same way that Cramer and Shoup do to achieve IND-CCA security. In fact, [7] require a *NIZK proof property and a computational property.* We only use a proof property of our HPS, and obtain secrecy from our assumption on the KEM, much like in the original NY paradigm. Hence we do *not* generalize the Cramer-Shoup approach to achieving IND-CCA secure encryption.

The technical assumption we use to capture a Diffie-Hellman type key exchange protocol is that of a *hard algebraic set system.* Roughly, an algebraic set system consists of a finite Abelian group S, together with a commutative, unitary sub-ring Φ of group endomorphisms over S that fulfil a number of natural algebraic properties. It is a hard algebraic set system if a Diffie-Hellman style computational problem is intractable. Examples of hard algebraic set systems can be obtained from standard computational assumptions such as the CDH and the RSA assumptions (using hardcore bit extraction). Our main result is an efficient transformation from any hard algebraic set system into a practical IND-CCA secure encryption scheme. With respect to the results our construction can be seen as a generalization of the recent specific constructions from computational problems [3,5,13,16].

1.1 Technical details

We now give some technical details of our transformation.

An IND-CPA secure construction. We start by describing a simple IND-CPA secure construction from any hard algebraic set system (S, Φ). It is actually a key encapsulation mechanism [8] (KEM) that can be viewed as a natural abstraction of the Diffie-Hellman key-exchange protocol. The scheme's secret key consists of a random $\chi \in \Phi$ and the public-key is a random $g \in S$ and $u = \chi(g) \in S$. Encryption picks random $\psi \in \Phi$, computes the ciphertext $c = \psi(g) \in S$ and uses the encapsulated key $K = \mathsf{Ext}(\psi(u))$ to blind the message. (Here Ext is an extractor function that is part of the underlying hard computational problem of the algebraic set system.) Decryption reconstructs the key by computing $K = \mathsf{Ext}(\chi(c))$. In our construction, we will have that χ and ψ commute. This directly implies correctness of the scheme, since then $\chi(c) = \chi(\psi(g)) = \psi(\chi(g)) = \psi(u)$.

Our IND-CCA secure construction. We augment the above IND-CPA secure construction in a clean and modular way (much like Naor and Yung) by adding a "replication part" and a "NIZK part" to the scheme. The two new parts require no computational assumptions, and so the resulting scheme is IND-CCA secure if the old scheme is IND-CPA secure. More concretely, ciphertexts are now tuples of the form $(c, \mathbf{d}, \pi)$, where c is from the IND-CPA construction, $\mathbf{d}$ is the "trapdoor element", and π is the "NIZK element" that proves consistency of the ciphertext. We now explain our construction by showing how the different parts affect the ability to perform decryption.

The idea behind the trapdoor element $\mathbf{d}$ in the ciphertext is that can be set up by a simulator such that it is possible to decrypt (without the knowledge of the scheme's secret key χ) all *consistent ciphertexts* $(c, \mathbf{d})$ except the ciphertext that is used to challenge the adversary (in the security reduction to the IND-CPA secure scheme). This "all-but-one" simulation technique can be traced back at least to [20], where it was used in the context of pseudorandom functions.[2] In the encryption context, "all-but-one" simulations have been used in identity-based encryption [2] and were already applied to several encryption schemes in [3,4,5,14,17,22,16].

The above all-but-one simulation technique allows to correctly simulate decryption of arbitrary for *consistent ciphertexts* $(c, \mathbf{d})$ but consistency can only be checked using the secret key which is not available during simulation. To provide an alternative consistency check we add the NIZK element π to the ciphertext. Actually, the NIZK element is generated using a hash proof system [7] and proves that $(c, \mathbf{d})$ is contained in the *trapdoor language* consisting of all consistent ciphertexts. However, we stress that we use hash proof system techniques here

[2] We stress that our use of the term "all-but-one" refers to the ability to generate a secret key that can be used to decrypt all consistent ciphertexts except for an *externally given* ciphertext. This is very different from the techniques of, e.g., [8]: in this latter framework, the first step in the proof consists in *making the challenge ciphertext inconsistent*, and then constructing a secret key that can be used to construct *all* consistent ciphertexts. Hence, "all-but-one" really refers to an "artificially punctured" secret key.

without relying on a (computational or decisional) assumption. Instead, we use a hash proof system only as a NIZK proof, in which case the hash proof system's soundness is information-theoretic.

We also note that the "trapdoor part" of a consistent ciphertext, along with the NIZK proof that the ciphertext is consistent, can be seen as a variant of an (extractable) NIZK proof of knowledge. However, in our case, the challenge ciphertext plays a special role: we need to construct the trapdoor language *from* a given challenge ciphertext. (Hence, extraction is—naturally—not possible for the challenge ciphertext.)

Our technical contribution (that may be of independent interest) is to bootstrap the trapdoor part and the the NIZK part (i.e., the hash proof system for the trapdoor language) generically from the abstract algebraic properties of algebraic set systems. In contrast to the generic NIZK-based constructions from [9,21] our constructions are relatively efficient: the key-size and ciphertexts of the obtained $\mathsf{IND\text{-}CCA}$ secure scheme contain $O(k)$ elements in S, where k is the security parameter. In many cases the ciphertexts can be "compactified" into a constant number of elements in S, giving truly practical schemes.

2 Preliminaries

2.1 Notation

Generic notation. A probabilistic polynomial-time (PPT) algorithm is a randomized algorithm which runs in strict polynomial time. By k we denote the security parameter, which indicates the "amount of security" we desire. A function $f : \mathbb{N} \to \mathbb{R}$ is negligible if for all $c \in \mathbb{N}$, there exists $k_0 \in \mathbb{N}$ such that $|f(k)| < k^{-c}$ for all $k > k_0$. Furthermore, f is overwhelming if $1 - f$ is negligible. For random variables X and Y, we write $X \overset{\mathrm{c}}{\approx} Y$ if X and Y are computationally indistinguishable, i.e., if for all PPT algorithms D, we have that $\Pr[\mathsf{D}(X) = 1] - \Pr[\mathsf{D}(Y) = 1]$ is negligible. Similarly, we write $X \overset{\mathrm{s}}{\approx} Y$ if the statistical distance between X and Y is negligible. For a vector $\mathbf{h} = (h_1, \ldots, h_\ell)$ and a nonempty set $J \subseteq \{1, \ldots, \ell\}$, we write $\mathbf{h}_J$ for the restricted vector $(h_i)_{i \in J}$. Furthermore, if ϕ is a function, then $\phi(\mathbf{h})$ denotes the component-wise application of ϕ, i.e., $\phi(\mathbf{h}) = (\phi(h_i))_i$.

Group endomorphisms. For an abelian group, we denote its group operation additively. If S is an abelian group, then $\mathrm{End}(S)$ consists of all group-homomorphisms $\chi : S \to S$. It has a ring-structure, where point-wise-addition is ring-addition (denoted "$+$") and functional composition is ring-multiplication (denoted "$\circ$"). Suppose $\varPhi$ is an additive sub-group of $\mathrm{End}(R)$. Then $\mathrm{Ann}(\varPhi) \subset S$, consists of all $g \in S$ for which $\chi(g) = 0$ for all $\chi \in \varPhi$, and it is a sub-group of S. A sub-ring of $\mathrm{End}(R)$ is unitary if it contains the identity endomorphism.

2.2 Key encapsulation mechanisms

Instead of a public-key encryption scheme we consider the conceptually simpler KEM framework. It is well-known that an $\mathsf{IND\text{-}CCA}$ secure KEM combined with a (one-time-)IND-CCA secure symmetric cipher (DEM) yields a IND-CCA secure public-key encryption scheme [8]. Efficient one-time IND-CCA secure DEMs can be constructed even without computational assumptions, using an encrypt-then-MAC paradigm [8], or using strong pseudorandom permutations.

Syntactics. A *key encapsulation mechanism (KEM)* $\mathsf{KEM} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ consists of three PPT algorithms. Via $(pk, sk) \leftarrow \mathsf{Gen}(1^k)$, the key generation algorithm produces public/secret keys for security parameter $k \in \mathbb{N}$; via $(K, C) \leftarrow \mathsf{Enc}(pk)$, the encapsulation algorithm creates a symmetric key[3] $K \in \{0,1\}^k$ together with a ciphertext C; via $K \leftarrow \mathsf{Dec}(sk, C)$, the possessor of secret key sk decrypts ciphertext C to get back a key K which is an element in $\{0,1\}^k$ or a special reject symbol $\perp$. For correctness, we require that for all possible $k \in \mathbb{N}$, and all $(K, C) \leftarrow \mathsf{Enc}(pk)$, we have $\Pr[\mathsf{Dec}(sk, C) = K] = 1$, where the probability is taken over the choice of $(pk, sk) \leftarrow \mathsf{Gen}(1^k)$, and the coins of all the algorithms in the expression above.

Security. The common requirement for a KEM is indistinguishability against chosen-ciphertext attacks (IND-CCA) [8], where an adversary is allowed to adaptively query a decapsulation oracle with ciphertexts to obtain the corresponding key. Formally:

Definition 1 **(IND-CCA security of a KEM).** *Let* $\mathsf{KEM} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be a KEM. For any PPT algorithm* A*, we define the following experiments* $\mathsf{Exp}^{\mathsf{CCA\text{-}real}}_{\mathsf{KEM},\mathsf{A}}$ *and* $\mathsf{Exp}^{\mathsf{CCA\text{-}rand}}_{\mathsf{KEM},\mathsf{A}}$*:*

$$\begin{array}{l|l}
\textbf{Experiment } \mathsf{Exp}^{\mathsf{CCA\text{-}real}}_{\mathsf{KEM},\mathsf{A}}(k) & \textbf{Experiment } \mathsf{Exp}^{\mathsf{CCA\text{-}rand}}_{\mathsf{KEM},\mathsf{A}}(k) \\
\quad (pk, sk) \leftarrow \mathsf{Gen}(1^k) & \quad (pk, sk) \leftarrow \mathsf{Gen}(1^k) \\
 & \quad R \leftarrow \{0,1\}^k \\
\quad (K^*, C^*) \leftarrow \mathsf{Enc}(pk) & \quad (K^*, C^*) \leftarrow \mathsf{Enc}(pk) \\
\quad \textit{Return } \mathsf{A}^{\mathsf{Dec}(sk,\cdot)}(pk, K^*, C^*) & \quad \textit{Return } \mathsf{A}^{\mathsf{Dec}(sk,\cdot)}(pk, R, C^*)
\end{array}$$

In the above experiments, the decryption oracle $\mathsf{Dec}(sk, C)$ *returns* $K \leftarrow \mathsf{Dec}(sk, C)$*, for all* $C \neq C^*$*. We define* A*'s advantage in breaking* KEM*'s* IND-CCA *security as*

$$\mathsf{Adv}^{\mathsf{CCA}}_{\mathsf{KEM},\mathsf{A}}(k) := \frac{1}{2}\left|\mathsf{Pr}\left[\mathsf{Exp}^{\mathsf{CCA\text{-}real}}_{\mathsf{KEM},\mathsf{A}}(k) = 1\right] - \mathsf{Pr}\left[\mathsf{Exp}^{\mathsf{CCA\text{-}rand}}_{\mathsf{KEM},\mathsf{A}}(k) = 1\right]\right|.$$

We say that KEM is IND-CCA secure *if* $\mathsf{Adv}^{\mathsf{CCA}}_{\mathsf{KEM},\mathsf{A}}$ *is negligible for all PPT* A.

As a stepping stone, we will also consider the weaker requirement of IND-CPA security of a KEM. The IND-CPA security experiment is very similar to the IND-CCA security experiment, only without a decryption oracle for the adversary:

Definition 2 **(IND-CPA security of a KEM).** *Let* $\mathsf{KEM} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be a KEM. For any PPT algorithm* A*, we define the following experiments* $\mathsf{Exp}^{\mathsf{CPA\text{-}real}}_{\mathsf{KEM},\mathsf{A}}$

[3] For simplicity we assume that the KEM's keyspace are bitstrings of length k.

and $\mathsf{Exp}^{\mathsf{CPA\text{-}rand}}_{\mathsf{KEM,A}}$ *as identical to the experiments* $\mathsf{Exp}^{\mathsf{CCA\text{-}real}}_{\mathsf{KEM,A}}$ *and* $\mathsf{Exp}^{\mathsf{CCA\text{-}rand}}_{\mathsf{KEM,A}}$ *from Definition 1, only that* A *does not get access to a decryption oracle* Dec*. Let*

$$\mathsf{Adv}^{\mathsf{CCA}}_{\mathsf{KEM,A}}(k) := \frac{1}{2}\left|\Pr\left[\mathsf{Exp}^{\mathsf{CPA\text{-}real}}_{\mathsf{KEM,A}}(k) = 1\right] - \Pr\left[\mathsf{Exp}^{\mathsf{CPA\text{-}rand}}_{\mathsf{KEM,A}}(k) = 1\right]\right|.$$

We say that KEM is $\mathsf{IND\text{-}CPA}$ secure *if* $\mathsf{Adv}^{\mathsf{CCA}}_{\mathsf{KEM,A}}$ *is negligible for all PPT* A.

3 Set Systems

3.1 Basic Definition

Definition 3 (Set system). *A* set system $\mathcal{SS} = (S, \Phi)$ *consists of the following*

- *A finite, non-empty set* S*.*
- *A non-empty set* Φ *of functions* $\chi : S \to S$*.*

Furthermore, we require that efficient algorithms exist for the following tasks:

- *Sampling* with the uniform distribution from* S*.*
- *Sampling* with the uniform distribution from* Φ*.*
- *Evaluating* $\chi(g)$ *when given* $\chi \in \Phi$ *and* $g \in S$*.*

Here, * *means that it is sufficient if sampling can be performed* approximatively uniform *(that is, if a distribution can be sampled which is statistically close to uniform).*

We stress that while our definitions are typically asymptotic, an explicit security parameter is sometimes suppressed for ease of exposition.

Definition 4 (Commutative set system). *A set system* (S, Φ) *is* commutative *if the functions in* Φ commute pairwise*, i.e., for all* $\chi, \psi \in \Phi$*, we have* $\chi \circ \psi = \psi \circ \chi$*.*

3.2 Hard Set Systems

The following definition encapsulates the computational hardness assumption associated with set systems.

Definition 5 (Hard set system). *Let* (S, Φ) *be a commutative set system, and let* $\mathsf{Ext} : S \to \{0,1\}^n$ *be efficiently computable. We say that* (S, Φ) *is a* hard set system with randomness extractor Ext *if*

$$(g, \chi(g), \psi(g), E) \quad \overset{\mathrm{c}}{\approx} \quad (g, \chi(g), \psi(g), R),$$

where $g \in S$*,* $\chi, \psi \in \Phi$*, and* $R \in \{0,1\}^n$ *are uniformly chosen, and* $E = \mathsf{Ext}(\chi(\psi(g))) \in \{0,1\}^n$.

3.3 Algebraic Set Systems

We now set abstract algebraic conditions that are sufficient for the existence of a quite efficient transformation that we will use to achieve CCA security.

Definition 6 (Algebraic set system). *A set system* (S, Φ) *is an* algebraic set system *if the following algebraic conditions are fulfilled.*
Group structure. *S is a finite Abelian group.*
Recognizability. *S is efficiently recognizable.*
Commutative endomorphisms. *Φ is a commutative, unitary sub-ring of* $\mathrm{End}(S)$.
Almost-transitivity. *A $g \in S$ is called* normal *if*

$$\forall h \in S \ \exists \phi \in \Phi : \ h = \phi(g) \ .$$

We require that a uniformly chosen $g \in S$ is normal with overwhelming probability.

Uniformity. *For uniformly chosen $g, u \in S$ and $\chi \in \Phi$, we have $(g, \chi(g)) \overset{s}{\approx} (g, u)$.*

Remark 1. If Φ consists of all multiplications by non-negative integers then Φ is a commutative, unitary sub-ring of $\mathrm{End}(S)$.

3.4 Examples

Diffie-Hellman. Let S be a cyclic group $\mathbb{G} = \langle g \rangle$ of prime order p. (For this and the next example, we stick to the more common notation and write the group multiplicatively.) We define Φ as

$$\Phi := \{\chi(g) = g^x \ : \ x \in \mathbb{Z}_p\}.$$

If we require that S is efficiently recognizable, this makes (S, Φ) a set system. Let $\chi, \psi \in \Phi$, i.e., $\chi(g) = g^x$ and $\psi(g) = g^y$, for some $x, y \in \mathbb{Z}_p$. Now $\chi(\psi(g)) = (g^y)^x = g^{xy} = (g^x)^y = \psi(\chi(g))$ and therefore (S, Φ) is commutative. Since S is efficiently recognizable, it is easy to see that (S, Φ) is also algebraic.

If the DDH assumption holds in $\mathbb{G}$, then (S, Φ) is a hard set system with randomness extractor $\mathsf{Ext} : \mathbb{G} \to \{0,1\}^n$, where Ext is an arbitrary pseudorandom generator. If the CDH assumption holds in $\mathbb{G}$, then (S, Φ) is a hard set system with randomness extractor $\mathsf{Ext}_s : \mathbb{G} \to \{0,1\}$. Here, Ext_s maps $g \in \mathbb{G}$ to the Goldreich-Levin bit $\sum_{i=1}^{|g|} g_i s_i$, where $|g|$ denotes the bit length and g_i the i-th bit of g in some canonical bit representation, and $s = (s_1, \ldots, s_{|g|}) \in \{0,1\}^{|g|}$.

We stress that knowledge of the *order* of $\mathbb{G}$ is not required. (Only one must be able to approximatively sample uniform exponents.) In particular, $\mathbb{G}$ could be instantiated over a higher-genus curve.

RSA. We use the group of *signed quadratic residues* [12,15]. Fix a Blum integer $N = PQ$ for safe primes $P, Q \equiv 3 \bmod 4$ (such that $P = 2p + 1$ and $Q = 2q + 1$ for primes p, q). Let $\mathbb{J}_N \subseteq \mathbb{Z}_N^*$ denote the set of elements with Jacobi symbol 1 modulo N and let $\mathbb{QR}_N \subset \mathbb{J}_N$ denote the set of quadratic residues modulo N. Consider the quotient group $S := \mathbb{QR}_N^+ := \mathbb{QR}_N/\pm 1$. Together with the group operation $a \circ b := |a \cdot b \bmod N|$ this forms a finite Abelian group of order pq. Furthermore, since $\mathbb{QR}_N^+ = \mathbb{J}_N^+ := \mathbb{J}_N/\pm 1 = \{|x| \; : \; x \in \mathbb{J}_N\}$, S is efficiently recognizable. Define

$$\Phi := \{\chi(g) = |g^x| \; : \; x \in \mathbb{Z}_{\lfloor N/4 \rfloor}\}.$$

Observe that we can sample uniformly from S and Φ. Furthermore, $(g, \chi(g))$ is statistically close to (g, u) for uniform $g, u \in S$ and $\chi \in \Phi$, since $\lfloor N/4 \rfloor$ approximates pq, the order of S, suitably well. This makes (S, Φ) a set system.

Finally, if the RSA assumption holds in $\mathbb{Z}_N$, then (S, Φ) is also hard with randomness extractor $\mathsf{Ext} : S \to \{0,1\}$, where Ext maps $g \in S$ to the least significant bit $\mathsf{LSB}(g)$ of g.

4 IND-CPA secure KEMs from commutative set systems

Construction 7 (Semantically secure KEM) *Assume that (S, Φ) is a hard commutative set system with randomness extractor* $\mathsf{Ext} : S \to \{0,1\}^n$. *Then, our basic key encapsulation scheme* $\mathsf{KEM} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$*, which is an obvious abstraction of the Diffie-Hellman scheme, is defined as follows.*

Key Generation. $\mathsf{Gen}(1^k)$ *chooses $g \in S$ and $\chi \in \Phi$ uniformly, and computes $u = \chi(g) \in S$. Public key is $pk = (g, u) \in S \times S$, and secret key is $sk = \chi \in \Phi$.*

Encapsulation. *Given $pk = (g, u) \in S \times S$,* Enc *chooses $\psi \in \Phi$ uniformly and computes the ciphertext $c = \psi(g) \in S$. Next,* Enc *derives the encapsulated key*

$$K = \mathsf{Ext}(\psi(u)) \in \{0,1\}^n. \tag{1}$$

Decapsulation. *Given $sk = \chi \in \Phi$ and $c \in S$,* Dec *computes*

$$\chi(c) = (\chi \circ \psi)(g) = (\psi \circ \chi)(g) = \psi(u)$$

to derive the encapsulated key $K \in \{0,1\}^n$ as in (1). Note that here it is exploited that the functions in Φ commute.

Theorem 1 (Construction 7 is an IND-CPA secure KEM). *If (S, Φ) is a hard commutative set system, then the KEM from Construction 7 is* $\mathsf{IND\text{-}CPA}$ *secure in the sense of Definition 2.*

Proof. This follows directly from Definition 5.

5 Hash proof systems

5.1 Definitions

We will use hash proof systems for a language L, as defined in [7, Section 5 of full version]. However, we stress that we will neither define nor use the concept of a subset membership problem (which essentially would require that elements in the language are computationally indistinguishable from elements outside of the language, see [7, Section 4 of full version]). For our purposes, only the proof system itself (whose security is defined information-theoretically) is relevant.

Definition 8 (Hash proof system). *Let L be a language and let ϵ be a real number with $0 \leq \epsilon < 1$. A hash proof system* with error probability ϵ *consists of the following.*

- *A finite non-empty set $\mathcal{V}$: this is where the verifier samples a secret verification-key from, to enable him to check proofs.*
- *A finite non-empty set $\mathcal{P}$ and a function $\alpha : \mathcal{V} \to \mathcal{P}$: this maps a verification key to its projection, which is an auxiliary input for the prover to construct a proof.*
- *A non-empty finite set Π: this is where proof strings will be sampled from.*

Furthermore, efficient algorithms for the following tasks exist.

- *Sampling with the uniform distribution from $\mathcal{V}$.*
- *Computing $\alpha(\kappa) \in \mathcal{P}$ when given $\kappa \in \mathcal{V}$.*
- *Computing the proof $\pi \in \Pi$ when given the statement $x \in L$, along with either the projection $\alpha(\kappa)$ and a witness $\phi \in \Phi$ (that $x \in L$), or, alternatively, the verification key κ itself.*

The following security properties hold, even in the presence of an unbounded adversary.

Completeness. *If indeed $x \in L$, a proof $\pi \in \Pi$ thus computed is* accepted *when verified using the secret verification key κ. This verification is performed efficiently by the verifier.*

Soundness. *For every $x \notin L$, every projection $P \in \mathcal{P}$, and every purported proof $\tilde{\pi} \in \Pi$: the probability (over uniform $V \in \mathcal{V}$ with $\alpha(V) = P$) that $\tilde{\pi}$ will be accepted is at most ϵ.*

Uniqueness. *The proof $\pi \in \Pi$ is unique. In the verification procedure referred to above, the verifier actually first computes π' from $x \in L$ and the verification key κ. The decision is then made by checking whether $\pi' = \pi$. In other words, the verifier can compute the proof himself from seeing the statement, using his secret verification key.*

Note that the uniqueness property implies a non-interactive zero-knowledge property, in the following sense. In the zero-knowledge setting, the verification key can be set up by a simulator, who then can generate the *unique* proofs π as $\pi = \kappa(x)$ for arbitrary statements x and without witness.

We make a number of remarks and comments concerning our definitions:

- The error probability ϵ can be decreased exponentially by running copies based on independently selected keys in parallel.
- Such a hash proof system will be "global" in the sense that it does not essentially depend on the length ℓ or on the choice of the base vectors $g, \mathbf{h}$. Furthermore, is assumed that the generation of the secret verification key does not depend on the choice of base vectors.
- Obviously, however, several technical details in the definition above will typically "scale with ℓ." Also, all algorithms involved may take ℓ and $g, \mathbf{h}$ as input (except secret key generation, which may not depend on $g, \mathbf{h}$, see above). But this dependence is suppressed in the notation.

5.2 Our trapdoor language

We define a natural language derived from set systems that simply "singles out" sequences of elements obtained by applying the same function to (a subset of) some fixed sequence elements. We note that [24] use the related but dual concept of "correlated products" to obtain chosen-ciphertext security. Namely, they apply several trapdoor functions to the same preimage, while in our approach, we apply one function to several preimages. We also note that in their work, it is crucial that the functions can be inverted (using a trapdoor). We do not have this requirement.

Definition 9 (Trapdoor language). *Let $(S, \varPhi)$ be a set system, let ℓ be a positive integer, and let*

$$g \in S, \quad \mathbf{h} = (h_1, \dots, h_\ell) \in S^\ell,$$

be base vectors. *Then the* trapdoor language L associated to $(S, \varPhi)$ and $g, \mathbf{h}$ *is defined as*

$$L = \{(c, \mathbf{d}, J) \in S \times S^J \times \mathcal{J} \mid \exists \chi \in \varPhi \textit{ such that } c = \chi(g) \wedge \mathbf{d} = \chi(\mathbf{h}_J)\},$$

where $\mathcal{J}$ consists of all non-empty subsets of $\{1, \dots, \ell\}$. (Recall our abbreviation $\chi(\mathbf{h}_J) = (\chi(h_i))_{i \in J}$.) Such a function $\chi \in \varPhi$ (not necessarily unique) is called a witness.

In the remaining part of this section we show the following theorem.

Theorem 2 (HPS for our trapdoor language). *Let $(S, \varPhi)$ be a an algebraic set system and let $g \in S$, $\mathbf{h} \in S^\ell$ be randomly chosen base vectors. If g is normal (in the sense of Definition 6) and $h_i \neq 0$ for all i, then there exists a hash proof system for the language L. The error probability is at most ℓ/p, where p is the smallest prime divisor of $|S|$.*

The proof proceeds in two steps. First we prove the case $\ell = 1$ and then we show how the general case follows from that by induction.

Let $g \in S$ be normal, and let $h \in S$. Since g is normal, $h = \rho(g)$ for some $\rho \in \varPhi$. We now construct a hash proof system for the trapdoor language L. The hash proof system is defined as follows.

$$Z = \{(\chi(g), \psi(h)) \ : \ \chi, \psi \in \varPhi\} \subset S \times S, \tag{2}$$

$$L = \{(\chi(g), \chi(h)) \ : \ \chi \in \varPhi\} \subset Z. \tag{3}$$

(For simplicity and ease of presentation, we omit the J component of Z and L, since in case $\ell = 1$ this component is trivial.)

Setup. The verifier chooses a random secret verification key $(\delta, \rho) \in \varPhi \times \varPhi$, and computes its projection

$$\alpha = \delta(g) + \rho(h).$$

Proof phase. The prover holds $(c,d) \in L$ and a witness $\chi \in \Phi$ such that

$$(c,d) = (\chi(g), \chi(h)).$$

He computes the proof

$$\pi = \chi(\alpha).$$

Verification. The verifier checks whether

$$\pi = \delta(c) + \rho(d).$$

Note that if the prover is honest, then indeed by commutativity

$$\pi = \chi(\alpha) = \chi(\delta(g) + \rho(h)) = \delta(\chi(g)) + \rho(\chi(h)) = \delta(c) + \rho(d)\ .$$

We sketch why the above hash proof system satisfies the conditions of Definition 8. The full proof (as well as the case of general ℓ) is contained in the full version of this paper [6]. Let $(c,d) \in Z$. Suppose the prover falsely claims that $(c,d) \in L$. The pair (δ, ρ) is randomly distributed on $\Phi \times \Phi$ conditioned on the projection being equal to α. Then, by a technical lemma, each solution z of the two equations $\alpha = \delta(g) + \rho(h)$ and $z = \delta(c) + \rho(d)$ is equally likely to be the "correct proof." Since there are at least p such solutions Theorem 2 now follows (for $\ell = 1$).

6 IND-CCA secure KEMs from algebraic set systems

Construction 10 (Chosen-ciphertext secure KEM) *Let (S, Φ) a hard algebraic set system with randomness extractor $\mathsf{Ext} : S \to \{0,1\}^n$. Further, assume a target collision resistant hash function T on S (whose formal definition can be looked in [6]). For $c \in S$, $\mathsf{T}(c)$ is encoded as a subset of $\{1, \ldots, 2k\}$, with $|\mathsf{T}(c)| = k$. Note that if $\mathsf{T}(c) \neq \mathsf{T}(c')$, then these two sets are incomparable by inclusion.*
Key generation. *Let (S, Φ) be an algebraic set system. Choose*

$$g \in S, \quad \mathbf{h} = (h_1, \ldots, h_{2k}) \in S^{2k}.$$

Using Theorem 2, set up an instance of the hash proof system from Section 5 (with negligible error probability ϵ) for the trapdoor language L, resulting in a verification key $\kappa \in \mathcal{V}$. Note that proofs for membership in L are from a set $\Pi \subseteq S^m$ for some m. Next, compute the projection value $\alpha = \alpha(\kappa) \in \mathcal{P}$. Finally, choose a function $\chi \in \Phi$ uniformly and compute

$$u = \chi(g) \in S.$$

The public/secret key pair is

$$pk = (g, u, \mathbf{h}, \alpha) \in S \times S \times S^{2k} \times \mathcal{P}, \quad sk = (\chi, \kappa) \in \Phi \times \mathcal{V}.$$

Encapsulation. *Given $pk = (g, u, \mathbf{h}, \alpha)$, choose a function $\psi \in \Phi$ at random, and compute*

$$c = \psi(g)$$

Next, compute $J = \mathsf{T}(c) \subset \{1, \ldots, 2k\}$ and

$$\mathbf{d} = \psi(\mathbf{h}_J) \in S^k$$

Using $\psi \in \Phi$, $\alpha \in \mathcal{P}$ and $d \in L$, compute the proof $\pi \in \Pi \subseteq S^m$ that $(c, \mathbf{d}, J) \in L$. The ciphertext consists of the pair

$$(c, \mathbf{d}, \pi) \in S \times S^k \times S^m,$$

and the session key is computed as

$$K = \mathsf{Ext}(\psi(u)) \in \{0,1\}^n. \tag{4}$$

Decapsulation. *Given $sk = (\chi, \kappa)$ and a ciphertext $(c, \mathbf{d}, \pi) \in S^{1+k+m}$, compute $J = \mathsf{T}(c) \subset \{1, \ldots, 2n\}$ and verify that $\pi \in S^m$ proves $(c, \mathbf{d}, J) \in L$. If the proof is invalid, reject. Otherwise, compute the session key as*

$$K = \mathsf{Ext}(\chi(c)) \in \{0,1\}^n.$$

Correctness. *We argue that the above KEM satisfies correctness. Note that for correctly generated ciphertexts, we have that*

$$(c, \mathbf{d}, \pi) = (\psi(g), \psi(\mathbf{h}_J), \pi),$$

where π is a proof that $(c, \mathbf{d}, J) \in L$. Hence, correctly generated ciphertexts are not rejected. Furthermore,

$$\chi(c) = (\chi \circ \psi)(g) = \psi(u),$$

which implies that decapsulation extracts the same key as encapsulation.

Theorem 3. *If (S, Φ) is a hard algebraic set system, then the above KEM is* IND-CCA *secure in the sense of Definition 1.*

Proof. We give a simulation of the IND-CCA experiment for an arbitrary PPT adversary A. It suffices to construct a simulator S such that the following holds. On input

$$(g, \chi(g), \psi(g), E^*)$$

(with g, χ, ψ, E^* as in Definition 5), S simulates the real IND-CCA experiment $\mathsf{Exp}^{\mathsf{CCA\text{-}real}}_{\mathsf{KEM},\mathsf{A}}$, and on input

$$(g, \chi(g), \psi(g), R^*),$$

S simulates the random IND-CCA experiment $\mathsf{Exp}^{\mathsf{CCA\text{-}rand}}_{\mathsf{KEM},\mathsf{A}}$.

Setup. So say that S is invoked on input (g, u, c^*, P), for $c^* = \psi(g)$, $u = \chi(g)$, and unknown $\chi, \psi \in \Phi$. Furthermore, $P \in \{0,1\}^n$ is either equal to the extraction E^* or random.

First, S sets up a substitute decapsulation key that can be used to decrypt all ciphertexts except the challenge ciphertext, which will be constructed around $\psi(g)$. Concretely, S computes from its own challenge (g, u, c^*, P) the value $J^* = \mathsf{T}(c^*) \subset \{1, \ldots, 2k\}$. Then, S chooses uniformly $\eta = (\eta_1, \ldots, \eta_{2k}) \in \varPhi$ and defines

$$h_i = \eta_i(g) \qquad \text{for } i \in J^*, \tag{5}$$

$$h_i = \eta_i(g) \cdot u \qquad \text{for } i \notin J^*. \tag{6}$$

Finally, S sets up a hash proof system for the trapdoor language L induced by g and $\mathbf{h}$ (see Definition 9). Let κ and α be the corresponding verification key and its projection. Then, S defines a public key pk along with a substitute secret key sk' as follows:

$$pk = (g, u, \mathbf{h}, \alpha) \in S \times S^\ell \times S^{2k} \times \mathcal{P} \qquad sk' = (\eta, \kappa) \in \varPhi^\ell \times \mathcal{V}.$$

Note that by the uniformity of $(S, \varPhi)$ (see Definition 6), the public keys prepared by S are statistically close to authentic public keys as produced by the key generation from Construction 10.

Challenge ciphertext and key. Next, S prepares a challenge ciphertext $(c^*, \mathbf{d}^*, \pi^*) \in S \times S^{J^*} \times S^m$. We have already defined c^* above, so it remains to define $\mathbf{d}^* = (d_i^*)_{i \in J^*}$ and π. Namely, S sets $d_i^* = \eta_i(c^*)$ for $i \in J^*$. Since

$$d_i^* = \eta_i(c^*) = \eta_i(\psi(g)) = \psi(\eta_i(g)) \overset{i \in J^*}{=} \psi(h_i),$$

this gives a $(c^*, \mathbf{d}^*)$ exactly as produced by the encapsulation algorithm of Construction 10. Because $(c^*, \mathbf{d}^*, J^*) \in L$, a proof π for that statement can be produced using the verification key κ. This yields a challenge ciphertext $(c^*, \mathbf{d}^*, \pi^*)$ exactly as produced by the encapsulation algorithm.

Note that if S's challenge P satisfies

$$P = E^* = \mathsf{Ext}((\chi \circ \psi)(g)),$$

then P equals the real key K as the encapsulation algorithm would have computed in (4), and hence P is distributed as the challenge key K in the real $\mathsf{IND\text{-}CCA}$ experiment $\mathsf{Exp}^{\mathsf{CCA\text{-}real}}_{\mathsf{KEM,A}}$. On the other hand, if P is random, then clearly P is distributed as a random challenge key in the $\mathsf{IND\text{-}CCA}$ experiment $\mathsf{Exp}^{\mathsf{CCA\text{-}rand}}_{\mathsf{KEM,A}}$.

Decapsulation queries. S then invokes adversary A with public key pk', challenge ciphertext $(c^*, \mathbf{d}^*, \pi^*)$, and challenge key P. By the above, this yields a view for A as in the real, resp. random $\mathsf{IND\text{-}CCA}$ experiment, depending on whether $P = E^*$ or P is random.

It remains to implement a decapsulation oracle for A. To this end, assume that A makes a decapsulation query $(c, \mathbf{d}, \pi)$. First, we may assume $c \in S$, $\mathbf{d} \in S^J$ (for $J = \mathsf{T}(c)$), and $\pi \in S^m$, since S is efficiently recognizable. If π is not a correct proof of $(c, \mathbf{d}, J) \in L$ according to κ, then S rejects, exactly as the authentic decapsulation algorithm would have done. In the following, we hence may further assume that π is a valid (with respect to verification key κ) proof that $(c, \mathbf{d}, J) \in L$. By the soundness of the hash proof system,[4] this in particular implies that, with overwhelming probability, there exists $\tilde{\psi} \in \Phi$ with $\tilde{\psi}(g) = c$ and $\tilde{\psi}(h_i) = d_i$ for all $i \in J$.

Observe that $c = c^*$ would imply $J^* = J$, so that for all $i \in J^* = J$,

$$d_i = \tilde{\psi}(h_i) = \eta_i(\tilde{\psi}(g)) = \eta_i(c) = \eta_i(c^*) = d_i^*.$$

By the uniqueness of valid proofs, this would hence imply $(c, \mathbf{d}, \pi) = (c^*, \mathbf{d}^*, \pi^*)$, which is a forbidden decapsulation query for A. Thus, we may even assume that $c \neq c^*$.

Without loss of generality, from $c \neq c^*$ it follows that $J = \mathsf{T}(c) \neq \mathsf{T}(c^*) = J^*$. (Otherwise, A has found a T-collision.) But $J \neq J^*$ implies that there exists an $i \in J \setminus J^*$, i.e., an $i \in J$ for which $h_i = \eta_i(g) \cdot u$. This allows S to derive $\chi(c)$ using

$$d_i = \tilde{\psi}(h_i) = \tilde{\psi}(\eta_i(g)) \cdot \tilde{\psi}(u) = \eta_i(\tilde{\psi}(g)) \cdot \chi(\tilde{\psi}(g)) = \eta_i(c) \cdot \chi(c)$$

and its knowledge about η_i. On the other hand, $\chi(c)$ allows to compute $K = \mathsf{Ext}(\chi(c))$ exactly as the decapsulation algorithm. Hence, the prepared substitute secret key $sk' = (\eta, \kappa)$ can be used to answer A's decapsulation queries.

Summarizing, the prepared simulation shows Theorem 3.

7 Discussion and variants

Global parameters. Note that the set system (S, Φ) employed in our encryption scheme can be re-used in many instances of the scheme. (In other words, there is no trapdoor related directly to the definition of (S, Φ) itself.) In particular, in the RSA set system from Section 3.4, no knowledge about the factorization of the modulus N is required. That means that N can be used as a *global* parameter for many parties.

Parallelization. In some of our examples from Section 3.4, the extracted values are only bits. This means that when implementing our generic CCA-secure encryption scheme with these examples, the corresponding KEM keys are only bits. However, it is possible to get larger keys by running several instances of the encryption scheme at once, without damaging the chosen-ciphertext security. Concretely, instead of publishing $u = \chi(g)$ in the public key, one can publish $u_i = \chi_i(g)$ for independently chosen $\chi_i \in \Phi$ ($i = 1, \ldots, n$). The sender still only uses one witness $\psi \in \Phi$ to compute $c = \psi(g)$, but now can extract from n separate values $\psi(u_1), \ldots, \psi(u_n)$. The adaptation of hash proof system and

[4] We stress that A only gets to see a proof π^* of a *valid* statement, which could have already been derived from the projected key α. Hence π^* does not disturb a reduction to the soundness of the hash proof system. This distinguishes our use of hash proof systems from the one in [8]. (In [8], the challenge ciphertext contains a proof of an *invalid* statement, which reveals information about the verification key κ beyond what is known from its projection α.)

trapdoor language are straightforward. (However, we stress that in order to decrypt, there must be $2k$ elements $h_{i,1}, \dots, h_{i,\ell}$ *for each* $i = 1, \dots, n$. Hence, not only the public key size, but also the ciphertext size grows linearly in n.)

Compact ciphertexts. For concrete set system platforms, we can substantially reduce the size of ciphertexts (from $O(k)$ group elements to $O(1)$). To see how, recall that in the $\mathsf{IND\text{-}CCA}$ secure encryption scheme, the ciphertext contains (the projection of) a vector $\mathbf{d} = \psi(\mathbf{h}_J)$, where $\mathbf{h}$ is part of the public key. The setup of $\mathbf{h}$ during the security proof (see (5)) has been chosen such that $\mathbf{d}$ allows to recover $\chi(c)$ as $\chi(c) = d_i / \eta_i(c)$ for any $i \in J \setminus J^*$. Now consider what happens if we substitute the vector $\mathbf{d}$ in the ciphertext with a single element

$$D := \psi\left(\prod_{i \in J} h_i\right) = \left(\prod_{i \in J} \eta_i(c)\right) \cdot \left(\prod_{i \in J \setminus J^*} \chi(c)\right).$$

Then, the simulation in the security proof can still derive $\prod_{i \in J \setminus J^*} \chi(c) = \chi(c)^{\Delta}$ for $\Delta := |J \setminus J^*|$. (Note that $0 < \Delta \le 2k$.) If we set $L := \mathrm{lcm}(1, \dots, 2k)$, then Δ divides L, so that the simulation can *always* compute $\psi(u)^L$. We can then modify the randomness extraction into $\mathsf{Ext}'(z) := \mathsf{Ext}(z^L)$, such that the decapsulation can be computed from $\chi(c)^L$ (instead of $\chi(c)$). Note that this automatically allows to compress the proof part π of the ciphertext down to one element. In particular, the ciphertext size (in group elements) is now *constant*. However, our modifications require that

$$(g, \chi(g), \psi(g), E') \quad \overset{c}{\approx} \quad (g, \chi(g), \psi(g), R), \tag{7}$$

where $g \in S$, $\chi, \psi \in \Phi$, and $R \in \{0,1\}^n$ are uniformly chosen, and $E' = \mathsf{Ext}'(\chi(\psi(g)) = \mathsf{Ext}(\chi(\psi(g))^L) \in \{0,1\}^n$. Note that (7) holds in the case of the Diffie-Hellman- and RSA-based set systems from Section 3.4 (since L and the order of S are coprime).

References

1. Daniel Bleichenbacher. Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1. In *CRYPTO'98*, volume 1462 of *LNCS*, pages 1–12. Springer, August 1998.
2. Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity based encryption without random oracles. In *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223–238. Springer, 2004.
3. Dan Boneh, Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. *SIAM Journal on Computing*, 36(5):915–942, 2006.
4. Xavier Boyen, Qixiang Mei, and Brent Waters. Direct chosen ciphertext security from identity-based techniques. In *ACM CCS 05*, pages 320–329. ACM Press, 2005.
5. David Cash, Eike Kiltz, and Victor Shoup. The twin Diffie-Hellman problem and applications. In *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 127–145. Springer, 2008.

6. Ronald Cramer, Dennis Hofheinz, and Eike Kiltz. Chosen-ciphertext Secure Encryption from Hard Algebraic Set Systems. Cryptology ePrint Archive, Report 2009/142.
7. Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 45–64. Springer, 2002.
8. Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003.
9. Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000.
10. Rosario Gennaro and Yehuda Lindell. A framework for password-based authenticated key exchange. *ACM Transactions on Information and System Security*, 9(2):181–234, 2006.
11. Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.
12. Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, April 1988.
13. Goichiro Hanaoka and Kaoru Kurosawa. Efficient chosen ciphertext secure public key encryption under the computational Diffie-Hellman assumption. In *ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 308–325. Springer, 2008.
14. Dennis Hofheinz and Eike Kiltz. Secure hybrid encryption from weakened key encapsulation. In *CRYPTO 2007*, volume 4622 of *LNCS*, pages 553–571. Springer, 2007.
15. Dennis Hofheinz and Eike Kiltz. The group of signed quadratic residues and applications. In *CRYPTO 2009*, volume 5677 of *LNCS*, pages 637–653. Springer, 2009.
16. Dennis Hofheinz and Eike Kiltz. Practical chosen ciphertext secure encryption from factoring. In *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 313–332. Springer, 2009.
17. Eike Kiltz. Chosen-ciphertext security from tag-based encryption. In *TCC 2006*, volume 3876 of *LNCS*, pages 581–600. Springer, 2006.
18. Eike Kiltz, Krzysztof Pietrzak, Martijn Stam, and Moti Yung. A new randomness extraction paradigm for hybrid encryption. In *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 590–609. Springer, 2009.
19. Kaoru Kurosawa and Yvo Desmedt. A new paradigm of hybrid encryption scheme. In *CRYPTO 2004*, volume 3152 of *LNCS*, pages 426–442. Springer, 2004.
20. Moni Naor, Omer Reingold, and Alon Rosen. Pseudo-random functions and factoring. *SIAM Journal on Computing*, 31(5):1383–1404, 2002.
21. Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd ACM STOC*. ACM Press, 1990.
22. Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In *40th ACM STOC*, pages 187–196. ACM Press, 2008.
23. Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *CRYPTO'91*, volume 576 of *LNCS*, pages 433–444. Springer, 1992.
24. Alon Rosen and Gil Segev. Chosen-ciphertext security via correlated products. In *TCC 2009*, volume 5444 of *LNCS*, pages 419–436. Springer, 2009.