A preliminary version of this paper appears in Advances in Cryptology – CRYPTO '07, Lecture Notes in Computer Science Vol. 4622, A. Menezes ed., Springer-Verlag, 2007. This is the full version.

Secure Hybrid Encryption from Weakened Key Encapsulation

Dennis Hofheinz   Eike Kiltz

CWI Amsterdam
The Netherlands
{hofheinz,kiltz}@cwi.nl

Abstract

We put forward a new paradigm for building hybrid encryption schemes from *constrained chosen-ciphertext secure* (CCCA) key-encapsulation mechanisms (KEMs) plus authenticated symmetric encryption. Constrained chosen-ciphertext security is a new security notion for KEMs that we propose. CCCA has less demanding security requirements than standard chosen-ciphertext (CCA) security (since it requires the adversary to have a certain plaintext-knowledge when making a decapsulation query) yet we can prove that CCCA is sufficient for secure hybrid encryption.

Our notion is not only useful to express the Kurosawa-Desmedt public-key encryption scheme and its generalizations to hash-proof systems in an abstract KEM/DEM security framework. It also has a very constructive appeal, which we demonstrate with a new encryption scheme whose security relies on a class of intractability assumptions that we show (in the generic group model) strictly weaker than the Decision Diffie-Hellman (DDH) assumption. This appears to be the first practical public-key encryption scheme in the literature from an algebraic assumption strictly weaker than DDH.

Keywords: Chosen-ciphertext security, weak security assumptions, hybrid encryption

1 Introduction

One of the main fields of interest in cryptography is the design and analysis of encryption schemes in the public-key setting (PKE schemes) that are secure against a very strong type of attacks — indistinguishability against chosen-ciphertext attacks (IND-CCA[1]) [30, 15]. In this work, we are interested in *practical schemes* with proofs of security under *reasonable security assumptions* (without relying on heuristics such as the random oracle model) and in *general methods* for constructing such schemes.

The first practical IND-CCA secure PKE scheme without random oracles was proposed in a seminal paper by Cramer and Shoup [12, 14]. Their construction was later generalized to hash proof systems [13]. In [36, 14] Cramer and Shoup also give a hybrid variant that encrypts messages of arbitrary length. The idea is to conceptually separate the key-encapsulation (KEM) part from the symmetric (DEM) part. Generally, this hybrid approach greatly improved

[1] In what follows IND-CCA always denotes the strong form of IND-CCA2 security.

practicality of encryption schemes. A folklore composition theorem (formalized in [14]) shows that if both KEM and DEM are CCA-secure then the hybrid encryption is CCA-secure. Common wisdom was that this sufficient condition was also necessary. However, at CRYPTO 2004, Kurosawa and Desmedt challenged this common wisdom by presenting a hybrid encryption scheme that demonstrates that a weaker security condition on the KEM may suffice for full CCA-secure hybrid encryption. Compared to the original Cramer-Shoup scheme, the scheme by Kurosawa and Desmedt improved efficiency and ciphertext expansion by replacing some of its algebraic components with *information theoretically* secure symmetric primitives. More recently, the KEM part of their scheme was indeed shown to be not CCA secure [18].

One natural open problem from [24] is if there exists a weaker yet natural security condition on the KEM such that, in combination with sufficiently strong symmetric encryption, chosen-ciphertext secure hybrid encryption can be guaranteed.

Extending the work of Cramer and Shoup [13], it was demonstrated in [24, 2, 17] that a variant of hash-proof systems (HPS) can be combined with symmetric encryption and a message authentication code (MAC) to obtain hybrid encryption. If the hash-proof system is *universal*$_2$, then the encryption scheme is chosen-ciphertext secure. However, the Kurosawa-Desmedt hybrid scheme could not be rigorously explained in this general HPS framework since the underlying hash-proof system is not universal$_2$. (Roughly, this is since universal$_2$ is a *statistical* property whereas the Kurosawa-Desmedt system contains a *computational* component, namely a target collision resistant (TCR) hash function.) In [24] (and [13]) only less efficient "hash-free variants" of their schemes could be explained through hash proof systems; security of all efficient TCR-based schemes had to be proved separately.

Surprisingly, almost all *practical* standard-model encryption schemes [12, 14, 24, 2, 11, 10, 22, 23] are based on the difficulty of Decision Diffie-Hellman (DDH) or stronger assumptions. This is contrasted by the existence of many natural groups in which the DDH assumption is known to be wrong; examples include pairing-groups and certain non prime-order groups like $\mathbb{Z}_p^*$. This often overlooked fact may turn into a serious problem in case DDH turns out to be wrong in all cryptographically interesting groups. In particular, [19] give evidence that groups with easy DDH problem, but hard computational Diffie-Hellman problem exist. [19] interpret this as an argument to rely on weaker assumptions than DDH.

1.1 Our contributions

A NEW KEM/DEM COMPOSITION THEOREM. We put forward the security notion of *indistinguishability against Constrained chosen-ciphertext attacks* (IND-CCCA) for KEMs which is stronger than IND-CPA (CPA stands for chosen-plaintext attacks) yet strictly weaker than IND-CCA. Intuitively, CCCA is separated from CCA security by only allowing an adversary to make a decapsulation query if it has sufficient "implicit knowledge" about the plaintext key to be decapsulated (hence the name "Constrained chosen-ciphertext security").[2]

As our main technical contribution we formalize the above notion and prove a composition theorem that shows that *any* IND-CCCA secure KEM combined with *any* authenticated (symmetric) encryption scheme yields IND-CCA secure hybrid encryption. This gives a positive answer to the open question from [24] mentioned before. Authenticated encryption is a quite general symmetric primitive and examples include "encrypt-then-mac" schemes (based on computationally secure primitives), and also more efficient single-pass schemes (see, e.g., [31]).

[2] This is reminiscent to the notion of "plaintext awareness" for public-key encryption [6] where it is infeasible for an adversary to come up with a valid ciphertext without being aware of the corresponding plaintext. Our definition is weaker in the sense that it only requires the adversary to have *implicit knowledge* on the plaintext.

Constrained chosen-ciphertext secure KEMs formalize a new design paradigm for efficient hybrid encryption. To guarantee chosen-ciphertext security for hybrid encryption schemes it is sufficient to verify a natural security condition on the key encapsulation part. We assess the constructive appeal of this framework by demonstrating that the original Kurosawa-Desmedt scheme [24], along with its variants [2, 29] and all hash-proof systems based schemes [13, 24], can be thoroughly explained through it. We furthermore present a new IND-CCCA secure KEM from the DDH assumption and show how to build a class of practical KEMs from progressively weaker assumptions than DDH.

Constrained chosen-ciphertext secure KEM from DDH. We propose a new KEM which is IND-CCCA secure under the DDH assumption. Although it relies on different proof techniques (it is not based on hash proof systems), syntactically it is reminiscent to the one by Kurosawa and Desmedt and can in fact be viewed as its *dual* (in the sense that certain parts from the ciphertext and the symmetric key are swapped in our scheme). Even though it is not much more efficient than the scheme by Kurosawa and Desmedt, we still consider it to be interesting since it constitutes the first efficient DDH-based encryption scheme that is not based on hash proof systems.

Constrained chosen-ciphertext secure KEM from n-Linear. Building on [9, 21] we introduce a new class of purely algebraic intractability assumptions, the n-Linear assumptions, where $n \geq 1$ is a parameter. They are such that the DDH assumption equals the 1-Linear assumption, the Linear assumption [9] equals the 2-Linear assumption, and the n-Linear assumptions become *strictly weaker* as the parameter n grows. More precisely, 1-Linear = DDH, and n-Linear implies $n + 1$-Linear, but (in the generic group model [35]) $n + 1$-Linear is still hard relative to an n-Linear oracle. In fact, for $n \geq 2$ the n-Linear assumption does not seem to be invalid in any obvious sense even in the groups from [19], in which the DDH problem is easy, and the computational Diffie-Hellman problem is supposedly hard. We generalize the KD scheme and its dual to a class of parametrized KEMs and prove their IND-CCCA security assuming n-Linear. These appear to be the first practical encryption schemes in the literature from a purely algebraic assumption which is strictly weaker than DDH.

Computational Hash-Proof Systems. We propose a purely computational variant of hash-proof systems. Generalizing [13, 24], we prove that computational hash-proof systems directly imply IND-CCCA secure KEMs. Hence, in combination with authenticated encryption, they yield efficient IND-CCA secure hybrid encryption. The Kurosawa-Desmedt scheme fits this framework, i.e. the underlying HPS is computational. This gives the first full explanation of the Kurosawa-Desmedt scheme in terms of HPS. As a generalization we provide computational hash-proof systems from the n-Linear assumptions hence explaining IND-CCCA security of our class of KEMs from the n-Linear assumptions.

1.2 Discussion and related work

In [1] (which is the full version of [2]), Abe et al. address the question from [24] about the existence of a natural weaker security condition for KEMs. They propose the notion of *LCCA secure KEMs with respect to the predicate* $\mathcal{P}^{\mathrm{mac}}$ and prove it sufficient to obtain, in combination with a MAC, IND-CCA secure tag-KEMs (and hence IND-CCA secure hybrid encryption). Though syntactically similar to ours, their notion *mingles* security of the KEM with the MAC part of the symmetric encryption scheme. The conceptual difference in our notion is that we give a general security definition for KEMs that is *completely independent* of any particular symmetric primitive. We think that this is more natural and more closely follows the spirit of the

KEM/DEM approach [14], where (for good reason) KEM and DEM are viewed as independent components.

Independent from this work Shacham [34] also proposes a family of hybrid encryption schemes from the n-Linear assumptions. His schemes can be viewed as a (slightly less efficient) Cramer-Shoup variant of our schemes from Section 5.2.

The 2-Linear assumption was introduced by Boneh, Boyen, and Shacham [9] and was later used in gap-groups to build an IND-CCA secure KEM [22]. For $n > 2$, Kiltz [21] introduced the class of *gap n-Linear* assumptions and (generalizing [22]) built a class of IND-CCA secure KEMs from it. Compared to n-Linear, in the latter gap-assumptions an adversary gets access to a DDH oracle which makes (for example) the gap 2-Linear assumption incomparable to DDH. In contrast, our motivation is to build schemes from an assumption weaker than DDH.

2 Preliminaries

2.1 Notation

If x is a string, then $|x|$ denotes its length, while if S is a set then $|S|$ denotes its size. If $k \in \mathbb{N}$ then 1^k denotes the string of k ones. If S is a set then $s \overset{\$}{\leftarrow} S$ denotes the operation of picking an element s of S uniformly at random. We write $\mathcal{A}(x, y, \ldots)$ to indicate that $\mathcal{A}$ is an algorithm with inputs $x, y, \ldots$ and by $z \overset{\$}{\leftarrow} \mathcal{A}(x, y, \ldots)$ we denote the operation of running $\mathcal{A}$ with inputs $(x, y, \ldots)$ and letting z be the output. We write $\mathcal{A}^{\mathcal{O}_1, \mathcal{O}_2, \ldots}(x, y, \ldots)$ to indicate that $\mathcal{A}$ is an algorithm with inputs $x, y, \ldots$ and access to oracles $\mathcal{O}_1, \mathcal{O}_2, \ldots$ and by $z \overset{\$}{\leftarrow} \mathcal{A}^{\mathcal{O}_1, \mathcal{O}_2, \ldots}(x, y, \ldots)$ we denote the operation of running $\mathcal{A}$ with inputs $(x, y, \ldots)$ and access to oracles $\mathcal{O}_1, \mathcal{O}_2, \ldots$, and letting z be the output.

2.2 Public-Key Encryption

A triple $\mathcal{PKE} = (\mathsf{PKE.kg}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ is a public-key encryption (PKE) scheme, if $\mathsf{PKE.kg}$ and $\mathsf{PKE.Enc}$ are probabilistic PTA, and $\mathsf{PKE.Dec}$ is a deterministic polynomial-time algorithm. For consistency, we require that for all $k \in \mathbb{N}$, all messages M, it must hold that $\Pr[\mathsf{PKE.Dec}(sk, \mathsf{PKE.Enc}(pk, M)) = M]$ is overwhelming in k, where the probability is taken over the above randomized algorithms and $(pk, sk) \overset{\$}{\leftarrow} \mathsf{PKE.kg}(1^k)$.

The security we require for PKE is IND-CCA security [30, 15]. To an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ we associate the following experiment $\mathbf{Exp}^{\mathrm{cca}}_{\mathcal{PKE},\mathcal{A}}(k)$.

```
**Experiment** $\mathbf{Exp}^{\mathrm{cca}}_{\mathcal{PKE},\mathcal{A}}(k)$
  $(pk, sk) \overset{\$}{\leftarrow} \mathsf{PKE.kg}(1^k)$
  $(M_0, M_1, St_1) \overset{\$}{\leftarrow} \mathcal{A}_1^{\mathsf{PKE.Dec}(sk,\cdot)}(pk)$ s.t. $|M_0| = |M_1|$
  $b \overset{\$}{\leftarrow} \{0,1\}$ ; $C^*_{pke} \overset{\$}{\leftarrow} \mathsf{PKE.Enc}(pk, M_b)$
  $b' \overset{\$}{\leftarrow} \mathcal{A}_2^{\mathsf{PKE.Dec}(sk,\cdot)}(C^*_{pke}, St_1)$
  If $b = b'$ return 1 else return 0
```

The adversary $\mathcal{A}_2$ is restricted not to query $\mathsf{PKE.Dec}(sk, \cdot)$ with C^*_{pke}. We define the advantage of $\mathcal{A}$ in the experiment as

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

PKE scheme $\mathcal{PKE}$ is said to be indistinguishable against chosen-ciphertext attacks (IND-CCA secure in short) if the advantage function $\mathbf{Adv}^{\mathrm{cca}}_{\mathcal{PKE},\mathcal{A}}(k)$ is a negligible function in k for all adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ with PTA $\mathcal{A}_1$, $\mathcal{A}_2$.

For integers k, t, Q we also define

$$\mathbf{Adv}^{\mathrm{cca}}_{\mathcal{PKE},t,Q}(k) \;=\; \max_{\mathcal{A}} \mathbf{Adv}^{\mathrm{cca}}_{\mathcal{PKE},\mathcal{A}}(k),$$

where the maximum is over all $\mathcal{A}$ that fulfill $t_{\mathcal{A}} \le t$ and $Q_{\mathcal{A}} \le Q$.

2.3 Key Encapsulation Mechanisms

A *key-encapsulation mechanism* $\mathcal{KEM} = (\mathsf{KEM.Kg}, \mathsf{KEM.Enc}, \mathsf{KEM.Dec})$ with key-space $\mathcal{K}(k)$ consists of three polynomial-time algorithms (PTAs). Via $(pk, sk) \overset{\$}{\leftarrow} \mathsf{KEM.Kg}(1^k)$ the randomized key-generation algorithm produces public/secret keys for security parameter $k \in \mathbb{N}$; via $(K, C) \overset{\$}{\leftarrow} \mathsf{KEM.Enc}(pk)$ the randomized encapsulation algorithm creates a uniformly distributed symmetric key $K \in \mathcal{K}(k)$ together with a ciphertext C; via $K \leftarrow \mathsf{KEM.Dec}(sk, C)$ the possessor of secret key sk decrypts ciphertext C to get back a key K which is an element in $\mathcal{K}$ or a special reject symbol $\perp$. For consistency, we require that for all $k \in \mathbb{N}$, and all $(K, C) \overset{\$}{\leftarrow} \mathsf{KEM.Enc}(pk)$ we have $\Pr[\,\mathsf{KEM.Dec}(sk, C) = K\,] = 1$, where the probability is taken over the choice of $(pk, sk) \overset{\$}{\leftarrow} \mathsf{KEM.Kg}(1^k)$, and the coins of all the algorithms in the expression above. Here we only consider only KEMs that produce perfectly uniformly distributed keys (i.e., we require that for all public keys pk that can be output by $\mathsf{KEM.Kg}$, the first component of $\mathsf{KEM.Enc}(pk)$ has uniform distribution).[3]

The common requirement for a KEM is indistinguishability against chosen-ciphertext attacks (IND-CCA) [14] where an adversary is allowed to adaptively query a decapsulation oracle with ciphertexts to obtain the corresponding key. We will not give the formal definition of IND-CCA for KEMs. Instead we refer the reader to Section 3 where we introduce a new, weaker security notion for KEMs that is sufficient for our goal of constructing IND-CCA secure hybrid encryption.

2.4 Authenticated Encryption

An authenticated symmetric encryption (AE) scheme $\mathcal{AE} = (\mathsf{AE.Enc}, \mathsf{AE.Dec})$ is specified by its encryption algorithm $\mathsf{AE.Enc}$ (encrypting $M \in \mathit{MsgSp}(k)$ with keys $K \in \mathcal{K}(k)$) and decryption algorithm $\mathsf{AE.Dec}$ (returning $M \in \mathit{MsgSp}(k)$ or $\perp$). Here we restrict ourselves to deterministic PTAs $\mathsf{AE.Enc}$ and $\mathsf{AE.Dec}$. The AE scheme needs to provide privacy (indistinguishability against one-time attacks) and authenticity (ciphertext authenticity against one-time attacks). This is *simulataneously* captured (similar to the more-time attack case [32]) by defining the ae-ot-advantage of an adversary $\mathcal{B}_{ae}$

$$\mathbf{Adv}^{ae\text{-}\mathrm{ot}}_{\mathcal{AE},\mathcal{B}_{ae}}(k) = \left|\Pr[K \overset{\$}{\leftarrow} \mathcal{K}(k)\ ;\ b \overset{\$}{\leftarrow} \{0,1\}\ ;\ b' \overset{\$}{\leftarrow} \mathcal{B}_{ae}^{\mathrm{LoR}_b(\cdot,\cdot),\mathrm{DoR}_b(\cdot)}(1^k)\ :\ b = b'] - 1/2\right| .$$

Here, $\mathrm{LoR}_b(M_0, M_1)$ returns $\psi \leftarrow \mathsf{AE.Enc}(K, M_b)$, and $\mathcal{B}_{ae}$ is allowed only one query to this left-or-right encryption oracle (one-time attack), with a pair of equal-length messages. Furthermore, the decrypt-or-reject oracle $\mathrm{DoR}_1(\psi)$ returns $M \leftarrow \mathsf{AE.Dec}(K, \psi)$ and $\mathrm{DoR}_0(\psi)$ always returns

[3] This requirement is met by all popular KEMs and makes our reduction in Theorem 3.1 tighter. However, we can show Theorem 3.1 also without this assumption, and derive that the keys are computationally close to uniform from our upcoming KEM security assumption. This comes at the price of a less tight security reduction in Theorem 3.1.

$\perp$ (reject), $\mathcal{B}_{ae}$ is allowed only one query to this decrypt-or-reject oracle which must be different from the output of the left-or-right oracle.

We say that $\mathcal{AE}$ is a *one-time secure authenticated encryption scheme* (AE-OT secure) if the advantage function $\mathbf{Adv}^{ae\text{-ot}}_{\mathcal{AE},\mathcal{B}_{ae}}(k)$ is negligible for all PTA $\mathcal{B}_{ae}$. Again, for integers k, t, $\mathbf{Adv}^{ae\text{-ot}}_{\mathcal{AE},t}(k) = \max_{\mathcal{B}_{ae}} \mathbf{Adv}^{ae\text{-ot}}_{\mathcal{AE},\mathcal{B}_{ae}}(k)$, where the maximum is over all $\mathcal{B}_{ae}$ that fulfill $t_{\mathcal{B}_{ae}} \leq t$.

3 Hybrid encryption from Constrained CCA secure KEMs

3.1 Constrained Chosen-Ciphertext Security for KEMs

The common requirement for a KEM is security against chosen-ciphertext attacks [14] where an adversary is allowed to adaptively query a decapsulation oracle with ciphertexts to obtain the corresponding key. We relax this notion to *contrained chosen-ciphertext security.* Intuitively, we only allow the adversary to make a decapsulation query if it already has some "a priori knowledge" about the decapsulated key. This partial knowledge about the key is modeled implicitly by letting the adversary additionally provide an efficiently computable Boolean predicate $\mathrm{pred} : \mathcal{K} \to \{0,1\}$. If $\mathrm{pred}(K) = 1$ then the decapsulated key K is returned, and $\perp$ otherwise. The amount of uncertainty the adversary has about the key (denoted as "plaintext uncertainty" where for KEMs the plaintext is the symmetric key) is measured by the fraction of keys the pedicate evaluates to 1. We require this fraction to be negligible, i.e. the adversary has to have a high a priori knowledge about the decapsulated key when making a decapsulation query.

We now turn to a more formal definition. To an adversary $\mathcal{A}$ we associate the following experiment $\mathbf{Exp}^{\mathrm{ccca}}_{\mathcal{KEM},\mathcal{A}}(k)$.

```
Experiment $\mathbf{Exp}^{\mathrm{ccca}}_{\mathcal{KEM},\mathcal{A}}(k)$
  $(pk, sk) \xleftarrow{\$} \mathsf{KEM.Kg}(1^k)$
  $K_0^* \xleftarrow{\$} \mathcal{K}(k)$ ; $(K_1^*, C^*) \xleftarrow{\$} \mathsf{KEM.Enc}(pk)$
  $b \xleftarrow{\$} \{0,1\}$
  $b' \xleftarrow{\$} \mathcal{A}^{\mathrm{CDec}(\cdot,\cdot)}(pk, K_b^*, C^*)$
  If $b = b'$ return 1 else return 0

$\mathrm{CDec}(\mathrm{pred}_i, C_i)$
  $K \leftarrow \mathsf{KEM.Dec}(sk, C_i)$
  If $K = \perp$ or $\mathrm{pred}_i(K) = 0$ then $\perp$
  Else return $K \in \mathcal{K}$
```

with the restriction that $\mathcal{A}$ is only allowed to query $\mathrm{CDec}(\mathrm{pred}_i, C_i)$ on predicates pred_i that are provided as PTA[4] and on ciphertexts C_i different from the challenge ciphertext C^*.

We define the advantage of $\mathcal{A}$ in the experiment as

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

For an adversary $\mathcal{A}$, let $t_{\mathcal{A}}$ denote the number of computational steps $\mathcal{A}$ runs (that includes the maximal time to *evaluate* each pred_i once), and let $Q_{\mathcal{A}}$ be the number of decapsulation queries $\mathcal{A}$ makes to its decapsulation oracle. For simplicity and without losing on generality, we consider only adversaries for which $t_{\mathcal{A}}$ and $Q_{\mathcal{A}}$ are independent of the environment that $\mathcal{A}$ runs in. To adversary $\mathcal{A}$ in the above experiment we also associate $\mathcal{A}$'s (implicit) plaintext uncertainty $\mathrm{uncert}_{\mathcal{A}}(k)$ when making decapsulation queries. Informally, $\mathrm{uncert}_{\mathcal{A}}(k)$ measures

[4] Technically, we charge the time required to evaluate each pred_i to $\mathcal{A}$'s runtime and require that $\mathcal{A}$ be polynomial-time.

the average fraction of keys that a predicate pred_i accepts, when running in environments that are at least as efficient as the original CCCA experiment.[5]

Formally, for an adversary $\mathcal{A}$ and an environment $\mathcal{E}$ that $\mathcal{A}$ interacts with (e.g., $\mathcal{E}$ could be the original CCCA experiment that interacts with $\mathcal{A}$), define

$$\mathrm{uncert}_{\mathcal{A},\mathcal{E}}(k) \;=\; \frac{1}{Q} \sum_{1 \le i \le Q} \Pr_{K \in \mathcal{K}}[\mathrm{pred}_i(K) = 1 \text{ when } \mathcal{A} \text{ runs with } \mathcal{E}] \;,$$

where $\mathrm{pred}_i : \mathcal{K} \to \{0,1\}$ is the predicate $\mathcal{A}$ submits in the ith decapsulation query. A CCCA adversary $\mathcal{A}$ is called *valid*, iff

- $\mathcal{A}$ is PTA, and
- for all environments $\mathcal{E}$ satisfying $t_{\mathcal{E}} \le t_{\mathrm{CCCA}}$, we have that $\mathrm{uncert}_{\mathcal{A},\mathcal{E}}(k)$ is negligible in k. Here, t_{CCCA} denotes the runtime of the original CCCA experiment (not counting the adversary runtime and the runtime taken for evaluating predicates).

Finally, a key encapsulation mechanism $\mathcal{KEM}$ is said to be *indistinguishable against constrained chosen ciphertext attacks* (IND-CCCA or simply CCCA) if for all valid PTA adversaries $\mathcal{A}$, the advantage function $\mathbf{Adv}^{\mathrm{ccca}}_{\mathcal{KEM},\mathcal{A}}(k)$ is negligible in k.

It is worth pointing out that by making different restrictions on $\mathrm{uncert}_{\mathcal{A}}(k)$ our notion of CCCA security leads to an interesting continuum between CPA and CCA security. With the restriction $\mathrm{uncert}_{\mathcal{A}}(k) = 0$ then CCCA = CPA; with the trivial restriction $\mathrm{uncert}_{\mathcal{A}}(k) \le 1$ (which makes is possible to always use the constant predicate $\mathrm{pred}(K) := 1$) then CCCA = CCA.

Concrete security. In the following, we will be interested in a concrete security treatment. That is, we want not only an asymptotic security statement from an asymptotic computational assumpion; we also want a statement that shows *exactly* how much security one gets from a given non-asymptotic version of the assumption.

First, the notion of a *valid* adversary is asymptotic and thus doesn't make sense in a concrete treatment. We refine the central notion of $\mathrm{uncert}(\cdot)$ therefore as follows:

$$\mathrm{uncert}_{\mathcal{A}}(k) \;=\; \max_{\substack{\mathcal{E} \\ t_{\mathcal{E}} \le t_{\mathrm{CCCA}}}} \frac{1}{Q} \sum_{1 \le i \le Q} \Pr_{K \in \mathcal{K}}[\mathrm{pred}_i(K) = 1 \text{ when } \mathcal{A} \text{ runs with } \mathcal{E}] \;,$$

where as before, t_{CCCA} denotes the runtime of the original IND-CCCA experiment. Note that we take the maximum of this average probability over all environments that are at least as efficient as the original IND-CCCA experiment.

Now the non-asymptotic, concrete version of CCCA security can be captured as follows: for integers k, t, Q, and for $0 \le \mu \le 1$, let

$$\mathbf{Adv}^{\mathrm{ccca}}_{\mathcal{KEM},t,Q,\mu}(k) \;=\; \max_{\substack{\mathcal{A} \\ t_{\mathcal{A}} \le t,\, Q_{\mathcal{A}} \le Q, \\ \mathrm{uncert}_{\mathcal{A}}(k) \le \mu}} \mathbf{Adv}^{\mathrm{ccca}}_{\mathcal{KEM},\mathcal{A}}(k),$$

We also stress the following: demanding that $\mathbf{Adv}^{\mathrm{ccca}}_{\mathcal{KEM},t,Q,\mu}(k)$ be negligible for all polynomials t, Q, and all negligible functions μ is *not* the same as demanding CCCA security. Namely,

[5] One might wonder why we require a certain property of $\mathcal{A}$'s submitted predicates even in more or less arbitrary environments $\mathcal{E}$ (instead of, say, only in the CCCA game). The reason is that to *show* CCCA security of a particular scheme, it will be helpful to use this stronger assumption on $\mathcal{A}$ in (slightly) modified environments (e.g., in an already slightly modified CCCA game).

the former is the *non-uniform* version of CCCA security. (That is, CCCA security against adversaries that are non-uniform, polynomial-sized circuit families.) This is not an artifact of our definition, but a general phenomenon of definitions geared towards capturing concrete security. In any case, all our reductions below consist of uniform reductions, so what we show is in fact both uniform and non-uniform security.

An alternative formulation of CCCA **security.** We remark that it is possible to restrict the CDEC$(\cdot,\cdot)$ oracle in the CCCA experiment to only output $\mathrm{pred}_i(K) \in \{0,1\}$ (and not the key itself in case $\mathrm{pred}_i(K) = 1$). Note that this does not restrict the adversary since in case $\mathrm{pred}_i(K) = 1$ it is always possible to reconstruct the whole key K by making $|\mathcal{K}(k)| = \mathrm{poly}(k)$ additional CCCA decapsulation queries with the predicates $\mathrm{pred}_{i,j}(K') :=$ "$\mathrm{pred}_i(K') \wedge \mathrm{bit}_j(K') = 1$", for $1 \le j \le |\mathcal{K}(k)|$. This determines the key K bit-wise.

3.2 Hybrid Encryption

Let $\mathcal{KEM} = (\mathsf{KEM.Kg}, \mathsf{KEM.Enc}, \mathsf{KEM.Dec})$ be a KEM and let $\mathcal{AE} = (\mathsf{AE.Enc}, \mathsf{AE.Dec})$ be an authenticated encryption scheme. We assume that the two schemes are compatible in the sense that for all security parameters k, we have that the KEM's and the AE's key-space are equal. Then we can consider a hybrid public key encryption scheme that encrypts arbitrary messages $M \in \mathit{MsgSp}$. The construction of $\mathcal{PKE} = (\mathsf{PKE.kg}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ is as follows.

```
$\mathsf{PKE.kg}(1^k)$
    $(pk, sk) \xleftarrow{\$} \mathsf{KEM.Kg}(1^k)$
    Return $(pk, sk)$
```

```
$\mathsf{PKE.Enc}(pk, M)$
    $(K, C) \xleftarrow{\$} \mathsf{KEM.Enc}(pk)$
    $\psi \leftarrow \mathsf{AE.Enc}(K, M)$
    Return $C_{pke} = (C, \psi)$
```

```
$\mathsf{PKE.Dec}(sk, C_{pke} = (C, \psi))$
    $K \leftarrow \mathsf{KEM.Dec}(sk, C)$
    $M \leftarrow \mathsf{AE.Dec}(K, \psi)$
    Return $M$ or $\perp$
```

Here $\mathsf{PKE.Dec}$ returns $\perp$ if either $\mathsf{KEM.Dec}$ or $\mathsf{AE.Dec}$ returns $\perp$.

The following shows that a IND-CCCA secure KEM and a AE-OT secure authenticated encryption scheme yields a IND-CCA secure PKE scheme.

Theorem 3.1 Assume $\mathcal{KEM}$ is secure in the sense of IND-CCCA and $\mathcal{AE}$ is secure in the sense of AE-OT. Then $\mathcal{PKE}$ is secure in the sense of IND-CCA. In particular,

$$\mathbf{Adv}^{\mathrm{cca}}_{\mathcal{PKE},t,Q}(k) \le \mathbf{Adv}^{\mathrm{ccca}}_{\mathcal{KEM},t,Q,2\mathbf{Adv}^{ae\text{-}\mathrm{ot}}_{\mathcal{AE},t'}(k),\mathcal{E}}(k) + (Q+1)\cdot \mathbf{Adv}^{ae\text{-}\mathrm{ot}}_{\mathcal{AE},t}(k) + \frac{Q}{|\mathcal{K}|}\ ,$$

where $t' := t + t_{\mathrm{CCCA}}$ for the runtime t_{CCCA} of the original IND-CCCA experiment.

The intuition of the proof is quite simple. The standard composition theorem [14] shows that in the above construction a IND-CCA secure KEM can be combined with a CCA secure DEM. Here we only require the KEM to be IND-CCCA secure. We deal with the full CCA decryption queries in the hybrid PKE scheme as follows. A decryption query of an adversary in the IND-CCA game consists of a KEM ciphertext C plus a DEM ciphertext ψ. In the reduction we use the predicate $\mathrm{pred}_\psi(\cdot)$ defined as $\mathrm{pred}_\psi(K) = 0$ if $\mathsf{AE.Dec}(K, \psi)$ returns $\perp$ and $\mathrm{pred}_\psi(K) = 1$ otherwise. (That is, ψ is hard-coded into pred_ψ.) By the ciphertext authenticity property of $\mathcal{AE}$ this predicate has small plaintext uncertainty, i.e. $\mathrm{uncert}(k) \le 2\mathbf{Adv}^{ae\text{-}\mathrm{ot}}_{\mathcal{AE},\mathcal{B}}(k)$. On the other hand, this hybrid decryption query can be correctly simulated using the output from the CCCA decapsulation query (which is a symmetric key or $\perp$) since an inconsistent ψ (with respect to the symmetric key) will already lead the predicate $\mathrm{pred}_\psi(\cdot)$ to be zero and hence the CCCA decapsulation query correctly returns reject. For a consistent ψ the predicate evaluates to one and the CCCA decapsulation query returns the correct symmetric key that in turn can be used to obtain the message from ψ.

We now give a formal proof of Theorem 3.1.

Proof: Let $\mathcal{A}$ be an adversary on the IND-CCA security of the hybrid scheme. We will consider a sequence of games, Game 1, Game 2, ..., each game involving $\mathcal{A}$. Let X_i be the event that in Game i, it holds that $b = b'$, i.e., that the adversary succeeds. We will make use of the following simple "Difference Lemma" [14].

Lemma 3.2 Let X_1,X_2, B be events, and suppose that $X_1 \wedge \neg B \Leftrightarrow X_2 \wedge \neg B$. Then $|\Pr[X_1] - \Pr[X_2]| \leq \Pr[B]$.

Game 1. The original PKE IND-CCA game, i.e. we have

$$|\Pr[X_1] - 1/2| = \mathbf{Adv}^{\mathrm{cca}}_{\mathcal{PKE},\mathcal{A}}(k) .$$

Game 2. Let $C^*_{pke} = (C^*, \psi^*)$ be the challenge ciphertext in the PKE IND-CCA game. In this game the decryption oracle in the first phase rejects all ciphertexts of the form $C_{pke} = (C^*, *)$. The view of adversary $\mathcal{A}$ is identical in Games 1 and 2 until a decryption query $(C^*, *)$ is made in the first phase of the IND-CCA experiment (so *before* $\mathcal{A}$ gets to see C^*).

Since the key K encapsulated in C^* is uniformly distributed and independent of $\mathcal{A}$'s view in the first phase, we have

$$|\Pr[X_2] - \Pr[X_1]| \leq \frac{Q}{|\mathcal{K}|} .$$

Note that each ciphertext uniquely determines a key.

Game 3. Replace the symmetric key K^* used to create the PKE challenge ciphertext with a random key K^*, uniformly independently chosen from $\mathcal{K}$. The proof of the following key lemma is postponed until later.

Lemma 3.3 $|\Pr[X_3] - \Pr[X_2]| \leq \mathbf{Adv}^{\mathrm{ccca}}_{\mathcal{KEM},t,Q,2\mathbf{Adv}^{ae\text{-}\mathrm{ot}}_{\mathcal{AE},t}(k)}(k)$.

Game 4. Reject all ciphertexts C_{pke} of the form $(C^*, *)$. Since ψ^* was generated using a random key $K^* \in \mathcal{K}$ that only leaks to $\mathcal{A}$ through ψ^*, authenticity of $\mathcal{AE}$ implies

$$|\Pr[X_4] - \Pr[X_3]| \ \leq \ Q_{\mathcal{A}} \cdot \mathbf{Adv}^{ae\text{-}\mathrm{ot}}_{\mathcal{AE},\mathcal{B}_{ae}}(k)$$

for a suitable adversary $\mathcal{B}_{ae}$ that simulates Game 3, using the LoR_b with two identical messages to obtain the AE part of the challenge ciphertext. $\mathcal{B}_{ae}$ simply uniformly picks one AE part of a decryption query of the form (C^*, ψ) to submit to the decrypt-or-reject oracle $\text{DoR}_1(\cdot)$.

Finally, Game 4 models one-time security of the AE scheme, and we have

$$|\Pr[X_4] - 1/2| \leq \mathbf{Adv}^{ae\text{-}\mathrm{ot}}_{\mathcal{AE},t}(k) .$$

Collecting the probabilities proves the theorem. ▌

It leaves to prove Lemma 3.3.

Proof of Lemma 3.3: We show that there exists an adversary $\mathcal{B}_{kem}$ against the IND-CCCA security of $\mathcal{KEM}$ with $t_{\mathcal{B}_{kem}} = t_{\mathcal{A}}$, $Q_{\mathcal{B}_{kem}} = Q_{\mathcal{A}}$, and for every PTA environment $\mathcal{E}$ there is an adversary $\mathcal{B}_{ae}$ against $\mathcal{AE}$ with $t_{\mathcal{B}_{ae}} = t_{\mathcal{A}} + t_{\mathcal{E}}$, such that

$$\begin{aligned} \mathrm{uncert}_{\mathcal{B}_{kem}}(k) &\leq 2\mathbf{Adv}^{ae\text{-ot}}_{\mathcal{AE},\mathcal{B}_{ae}}(k) && (1)\\ \Pr[X_2] &= \Pr[\mathbf{Exp}^{\mathrm{ccca}}_{\mathcal{KEM},\mathcal{B}_{kem}}(k) = 1 \mid b = 1] && (2)\\ \Pr[X_3] &= \Pr[\mathbf{Exp}^{\mathrm{ccca}}_{\mathcal{KEM},\mathcal{B}_{kem}}(k) = 1 \mid b = 0]\ . && (3) \end{aligned}$$

The adversary $\mathcal{B}_{kem}$ against the CCCA security of $\mathcal{KEM}$ is defined as follows. $\mathcal{B}_{kem}$ inputs (pk, K_b^*, C^*) for an unknown bit b. First, $\mathcal{B}_{kem}$ runs $\mathcal{A}_1$ on input pk. For the ith decryption query (C_i, ψ_i) made by adversary $\mathcal{A}_1$, adversary $\mathcal{B}_{kem}$ defines the function $\mathrm{pred}_i : \mathcal{K} \to \{0,1\}$ as

$$\mathrm{pred}_i(K) := \begin{cases} 0 & : \text{ if } \mathsf{AE.Dec}(K, \psi_i) \text{ returns } \perp \\ 1 & : \text{ otherwise} \end{cases}$$

Note that the symmetric ciphertext ψ_i is hard-coded into $\mathrm{pred}_i(\cdot)$. Clearly, $\mathrm{pred}_i(\cdot)$ is efficiently computable. If $C_i = C^*$ then $\mathcal{B}$ returns $\perp$. Otherwise, $\mathcal{B}_{kem}$ queries (pred_i, C_i) to its own oracle $\mathrm{CDec}(\cdot,\cdot)$ and receives the following answer: If $\mathsf{KEM.Dec}(sk, C_i)$ returns a key $K_i \in \mathcal{K}$ such that $\mathsf{AE.Dec}(K_i, \psi_i) \neq \perp$, then $\mathrm{CDec}(\mathrm{pred}_i, C_i)$ returns the key K_i. Otherwise (if $\mathsf{KEM.Dec}(sk, C_i) = \perp$ or if $\mathsf{AE.Dec}(K_i, \psi_i) = \perp$), $\mathrm{CDec}(\mathrm{pred}_i, C_i)$ returns $\perp$. Note that by the syntax of $\mathcal{AE}$ this can be used to perfectly simulate $\mathcal{A}$'s decryption queries.

For $\mathcal{A}$'s encryption challenge for two messages M_0, M_1, $\mathcal{B}_{kem}$ uses its own input (K_b^*, C^*) together with a random bit δ to create a challenge ciphertext $C^*_{pke} = (C^*, \psi^* \leftarrow \mathsf{AE.Enc}(K_b^*, M_\delta))$ of message M_δ. Adversary $\mathcal{B}_{kem}$ runs $\mathcal{A}_2(C^*_{pke}, St_1)$, answering decryption queries as defined above with the difference that all decryption queries of the form (C^*, ψ) (with $\psi \neq \psi^*$) are answered with withever $\mathsf{AE.Dec}(K_b^*, \psi)$ returns (a message or $\perp$). Evntually, $\mathcal{A}_2$ returns a guess bit δ' for δ and $\mathcal{B}_{kem}$ concludes its game with outputting $b' = 1$ if $\delta = \delta'$ and $b' = 0$, otherwise. This completes the description of $\mathcal{B}_{kem}$.

Adversary $\mathcal{B}_{kem}$ always perfectly simulates $\mathcal{A}$'s decapsulation queries. In case $b = 1$, $\mathcal{B}_{kem}$ uses the real key K_1^* for $\mathcal{A}$'s simulation which implies Equation (2). In case $b = 0$, $\mathcal{B}_{kem}$ uses a random key K_0^* for $\mathcal{A}$'s simulation which implies Equation (3).

The complexity bounds for $\mathcal{B}_{kem}$ are clear from the construction, and it is left to show that for any given efficient environment $\mathcal{E}$, $\mathrm{uncert}_{\mathcal{B}_{kem}}(k) = 2\mathbf{Adv}^{ae\text{-ot}}_{\mathcal{AE},\mathcal{B}_{ae}}(k)$ for a suitable $\mathcal{B}_{ae}$.

To this end we build an adversary $\mathcal{B}_{ae}$ against the AE security of $\mathcal{AE}$ as follows. $\mathcal{B}_{ae}$ inputs 1^k and internally simulates an interaction between $\mathcal{A}$ and $\mathcal{E}$ completely faithfully. However, $\mathcal{B}_{ae}$ additionally picks a random index $j^* \in \{1, \ldots, Q\}$. On $\mathcal{A}$'s j^* decryption query (C_{j^*}, ψ_{j^*}), $\mathcal{B}_{ae}$ submits ψ_{j^*} to its own decryption-or-reject oracle $\mathrm{DoR}_b(\cdot)$, and outputs $b' = 0$ iff $\mathrm{DoR}_b(\cdot)$ rejects with $\perp$.

Now $\mathcal{B}_{ae}$ will always output $b' = 0$ if $b = 0$ by definition of DoR_0. In case $b = 1$, $\mathcal{B}_{ae}$ will output $b' = 1$ iff the ciphertext ψ_{j^*} is valid in the sense $\mathsf{AE.Dec}(K', \psi_{j^*}) \neq \perp$ for an independent, uniformly (by the AE experiment) chosen key K'. So adversary $\mathcal{B}_{ae}$'s advantage is as follows.

$$\mathbf{Adv}^{ae\text{-ot}}_{\mathcal{AE},\mathcal{B}_{ae}}(k) = \frac{1}{2}\Pr[K' \xleftarrow{\$} \mathcal{K} : \mathsf{AE.Dec}(K', \psi_{j^*}) \neq \perp] = \frac{1}{2Q_{\mathcal{A}}}\sum_{j^*=1}^{Q_{\mathcal{A}}} \mathrm{pred}_{j^*}(K') = \frac{1}{2}\mathrm{uncert}_{\mathcal{B}_{kem},\mathcal{E}}(k),$$

where $\mathrm{pred}_{j^*}(\cdot) = \mathsf{AE.Dec}(\cdot, \psi_{j^*})$ is the predicate adversary $\mathcal{B}_{kem}$ submits to oracle CDec as the j^*th query. ▌

4 Efficient Key Encapsulation from DDH

4.1 Building blocks

We describe the building blocks used and assumptions made about them.

Group schemes. A group scheme $\mathcal{GS}$ [14] specifies a sequence $(\mathcal{GR}_k)_{k\in\mathbb{N}}$ of group descriptions. For every value of a security parameter $k \in \mathbb{N}$, $\mathcal{GR}_k$ specifies the four tuple $\mathcal{GR}_k = (\hat{\mathbb{G}}_k, \mathbb{G}_k, p_k, g_k)$ (for notational convenience we sometimes drop the index k). $\mathcal{GR}_k = (\hat{\mathbb{G}}, \mathbb{G}, p, g)$ specifies a finite abelian group $\hat{\mathbb{G}}$, along with a prime-order subgroup $\mathbb{G}$, a generator g of $\mathbb{G}$, and the order p of $\mathbb{G}$. We denote the identity element of $\mathbb{G}$ as $1_{\mathbb{G}} \in \mathbb{G}$. We assume that $\hat{\mathbb{G}}$ is of order $q = p'p$ and that it takes $|q|$ bits to represent an element in $\mathbb{G}$. We further assume the existence of an efficient sampling algorithm $x \stackrel{\$}{\leftarrow} \mathbb{G}$ and an efficient membership algorithm that test if a given element $x \in \hat{\mathbb{G}}$ is contained in the subgroup $\mathbb{G}$.

We further assume the DDH problem is hard in $\mathcal{GS}$, captured by defining the ddh-advantage of an adversary $\mathcal{B}_{\mathrm{ddh}}$ as

$$\mathbf{Adv}^{\mathrm{ddh}}_{\mathcal{GS},\mathcal{B}_{\mathrm{ddh}}}(k) = \frac{1}{2} \left|\Pr[\mathcal{B}_{\mathrm{ddh}}(g, h, g^a, h^a) = 1] - \Pr[\mathcal{B}_{\mathrm{ddh}}(g, h, g^a, K) = 1]\right|,$$

where g, $h, K \stackrel{\$}{\leftarrow} \mathbb{G}$ and $a \leftarrow \mathbb{Z}_p^*$.

Authenticated Encryption. We need an abstract notion of algebraic authenticated encryption where the keyspace consists of $\mathbb{G}$, secure in the sense of OT-AE. In Appendix D we recall (following the encrypt-then-mac approach [5, 14]) how to build such algebraic AE satisfying all required functionality and security from the following basic primitives:

- A (computationally secure) one-time symmetric encryption scheme with binary k-bit keys (such as AES or padding with a PRNG)
- A (computationally secure) MAC (existentially unforgeable) with k-bit keys
- A (computationally secure) key-derivation function (pseudorandom).

We remark that for our purposes it is also possible to use a more efficient single-pass authenticated encryption scheme (see, e.g., [31]). In both cases the the ciphertext expansion (i.e., ciphertext size minus plaintext size) of the AE scheme is only k (security parameter) bits which is optimal with respect to our security notion.

Target Collision Resistant Hashing. $\mathcal{TCR} = (\mathsf{TCR}_k)_{k\in\mathbb{N}}$ is a family of keyed hash functions $\mathsf{TCR}_k^s : \mathbb{G} \to \mathbb{Z}_p$ for each k-bit key s. It is assumed to be target collision resistant (TCR) [14], which is captured by defining the tcr-advantage of an adversary $\mathcal{B}_{\mathrm{tcr}}$ as

$$\mathbf{Adv}^{\mathrm{tcr}}_{\mathsf{TCR},\mathcal{B}_{\mathrm{tcr}}}(k) = \Pr[\mathsf{TCR}^s(c^*) = \mathsf{TCR}^s(c) \wedge c \neq c^* \ : \ s \stackrel{\$}{\leftarrow} \{0,1\}^k \ ; \ c^* \stackrel{\$}{\leftarrow} \mathbb{G} \ ; \ c \stackrel{\$}{\leftarrow} \mathcal{B}_{\mathrm{tcr}}(s, c^*)].$$

Note TCR is a weaker requirement than collision-resistance, so that, in particular, any practical collision-resistant function can be used. Also note that our notion of TCR is related to the stronger notion of universal one-way hashing [25], where in the security experiment of the latter the target value c^* is chosen by the adversary (but before seeing the hash key s).

Commonly [14, 24] this function is implemented using a dedicated cryptographic hash function like MD5 or SHA, which we assume to be target collision resistant. Since $|\mathbb{G}| = |\mathbb{Z}_p| = p$ we can alternatively also use a fixed (non-keyed) bijective encoding function $\mathsf{INJ} : \mathbb{G} \to \mathbb{Z}_p$. In that case we have a perfectly collision resistant hash function, i.e. $\mathbf{Adv}^{\mathrm{tcr}}_{\mathsf{INJ},\mathcal{B}_{\mathrm{tcr}}}(k) = 0$. In Appendix C, we show how to build such bijective encodings for a number of concrete group schemes.

4.2 The key-encapsulation mechanism

Let $\mathcal{GS}$ be a group scheme where $\mathcal{GR}_k$ specifies $(\hat{\mathbb{G}}, \mathbb{G}, g, p)$ and let $\mathsf{TCR} : \mathbb{G} \to \mathbb{Z}_p$ be a target collision resistant hash function (for simplicity we assume TCR to be non-keyed). We build a key encapsulation mechanism $\mathcal{KEM} = (\mathsf{KEM.kg}, \mathsf{KEM.Enc}, \mathsf{KEM.Dec})$ with $\mathcal{K} = \mathbb{G}$ as follows.

$\mathsf{KEM.Kg}(1^k)$	$\mathsf{KEM.Enc}(pk)$	$\mathsf{KEM.Dec}(sk, C)$
$x, y, \omega \xleftarrow{\$} \mathbb{Z}_p^*$	$r \xleftarrow{\$} \mathbb{Z}_p^*$; $c \leftarrow g^r$	Parse C as $(c, \pi) \in \hat{\mathbb{G}} \times \hat{\mathbb{G}}$
$u \leftarrow g^x$; $v \leftarrow g^y$; $h \leftarrow g^\omega$	$t \leftarrow \mathsf{TCR}(c)$; $\pi \leftarrow (u^t v)^r$	if $c \notin \mathbb{G}$ return $\perp$
$pk \leftarrow (u, v, h) \in \mathbb{G}^3$	$C \leftarrow (c, \pi) \in \mathbb{G}^2$	$t \leftarrow \mathsf{TCR}(c)$
$sk \leftarrow (x, y, \omega) \in (\mathbb{Z}_p)^3$	$K \leftarrow h^r \in \mathbb{G}$	if $c^{xt+y} \neq \pi$ return $\perp$
Return (sk, pk)	Return (C, K)	Return $K \leftarrow c^\omega$

We stress that decryption never explicitly checks if $\pi \in \mathbb{G}$; this check happens implicitly when $c \in \mathbb{G}$ and $c^{xt+y} = \pi$ is checked.

A correctly generated ciphertext has the form $C = (c, \pi) \in \mathbb{G} \times \mathbb{G}$, where $c = g^r$ and $\pi = (u^t v)^r = (g^{xt+y})^r = c^{xt+y}$. Hence decapsulation will not reject and compute the key $K = c^\omega = h^r$, as in encapsulation.

Encryption takes four standard exponentiations plus one application of TCR, where the generation of π can also be carried out as a single multi-exponentiation [7]. Decryption takes two exponentiations plus one application of TCR, where the two exponentiations can also be viewed as one sequential exponentiation [7] (which is as efficient as a multi-exponentiation) to simultaneously compute c^{xt+y} and c^ω.

Theorem 4.1 Let $\mathcal{GS}$ be a group scheme where the DDH problem is hard and assume $\mathcal{TCR}$ is target collision resistant. Then $\mathcal{KEM}$ is secure in the sense of IND-CCCA. In particular,

$$\mathbf{Adv}^{\mathrm{ccca}}_{\mathcal{KEM},t,Q,\mathrm{uncert}(k)}(k) \leq \mathbf{Adv}^{\mathrm{ddh}}_{\mathcal{GS},t}(k) + \mathbf{Adv}^{\mathrm{tcr}}_{\mathcal{TCR},t}(k) + Q \cdot \mathrm{uncert}(k) + \frac{Q}{p}.$$

In combination with Theorem 3.1 we obtain the following concrete security result.

Corollary 4.2 The hybrid encryption scheme $\mathcal{PKE}$ obtained by combining $\mathcal{KEM}$ with authenticated encryption $\mathcal{AE}$ is secure in the sense of IND-CCA. In particular,

$$\mathbf{Adv}^{\mathrm{cca}}_{\mathcal{PKE},t,Q}(k) \leq \mathbf{Adv}^{\mathrm{ddh}}_{\mathcal{GS},t}(k) + \mathbf{Adv}^{\mathrm{tcr}}_{\mathcal{TCR},t}(k) + (2Q+1) \cdot \mathbf{Adv}^{ae\text{-}\mathrm{ot}}_{\mathcal{AE},t}(k) + \frac{2Q}{p}\ .$$

Before we give a formal proof we give some intuition why the KEM is IND-CCCA secure. The difficulty with the simulation is that an adversary against the DDH assumption (simulating an adversary's view) has to distinguish between consistent ciphertexts (i.e., ciphertexts for that $c^{xt+y} = \pi$ holds) and inconsistent ciphertexts, without knowing the secret key. The idea of the proof is as follows. The simulator inputs $(g, h, c^* = g^r, K^*)$ and wants to distinguish $K^* = h^r$ from a random element in $\mathbb{G}$. In the simulation the values u, v from the public-key are set-up such that the tuple (c^*, π^*) can be used as the challenge ciphertext (for some efficiently computable π^*) and the value K^* as the session key. By construction, the corresponding real session key is h^r so breaking IND of the KEM is equivalent to solving the DDH problem. It leaves to deal with the decapsulation queries under a CCCA attack. The simulator is not able to distinguish consistent from inconsistent ciphertexts. However, the simualor uses an alternative decapsulation algorithm with the following two properties:

- If the queried ciphertext is *consistent* (and as long as it is distinct from the challenge ciphertext) then the alternative decapsulation algorithm yields the correct session key K. This is done using an algebraic trick from selective-ID secure identity-based encryption [8].
- If the queried ciphertext is *inconsistent* then the alternative decapsulation algorithm yields one virtual session key K that is uniformly distributed over $\mathbb{G}$ (in an information theoretic sense). The probability space is taken over all possible secret keys of the simulator that yield the public-key given to the adversary. Returning the virtual key K to the adversary would completely determine the simulator's secret key and hence also the virtual key K' for the next decapsulation query. However, in the IND-CCCA game it will be hard for an adversary to provide sufficient information about K (in form of the predicate pred) such that inconsistent decapsulation queries will nearly always lead to a rejection and the same argument can be repeated iteratively.

We now turn to a formal proof.

Proof: First, if a key pair (pk, sk) with $pk = (h, u, v)$ and $sk = (\omega, x, y)$ is clear from the context, we call a ciphertext $C = (c, \pi)$ *consistent* iff $c^{xt+y} = \pi$ holds for the tag $t := \mathsf{TCR}(c)$. Note that C is hence consistent iff $\log_g(c) = \log_{u^t v}(\pi)$.

Let $\mathcal{A}$ be an adversary on the IND-CCCA security of the KEM. We will consider a sequence of games, Game 1, Game 2, ..., each game involving $\mathcal{A}$. Let X_i be the event that in Game i, it holds that $b = b'$, i.e., that the adversary succeeds.

Game 1. The KEM IND-CCCA game with random $b \in \{0, 1\}$, i.e., we have

$$|\Pr[X_1] - 1/2| = \mathbf{Adv}^{\text{ccca}}_{\mathcal{KEM},\mathcal{A}}(k) .$$

Let us fix some notation. Let $C^* = (c^*, \pi^*) = (g^a, (u^{t^*} v)^a)$ be the challenge ciphertext (where $t^* = \mathsf{TCR}(c^*)$) and let $K_1^* = h^a$ be the real challenge key.

Game 2. The decryption oracle immediately rejects all ciphertexts (c, π) with $c \neq c^*$ and $t = t^*$ (TCR check). Since Game 1 and Game 2 proceed identically until $c \neq c^*$ and $\mathsf{TCR}(c) = t = t^* = \mathsf{TCR}(c^*)$, we have Lemma 3.2

$$|\Pr[X_2] - \Pr[X_1]| \leq \mathbf{Adv}^{\text{tcr}}_{\mathsf{TCR},\mathcal{B}_{\text{tcr}}}(k).$$

Game 3. Change generation of the secret key as follows. Pick uniformly values $x_1, x_2, y_1 \in \mathbb{Z}_p$ with $x_2 \neq 0$ and define

$$x = x_1 + \omega x_2 \; ; \; y = y_1 + \omega(-t^* x_2) . \tag{4}$$

Note that public and secret key have exactly the same distribution as in the last game. We will now rewrite the experiment in terms of x_1, x_2, and y_1 (our goal is to run the experiment without knowledge of ω). Equation (4) defines the public key as

$$u = g^{x_1} h^{x_2} \; ; \; v = g^{y_1} h^{-t^* x_2} \tag{5}$$

Note that now the consistency check $c^{xt+y} = \pi$ needs to know ω. The change of the secret key also implicitly affects the generation of the element π^* in the challenge ciphertext. Creation of challenge ciphertext C^* and real key K_1^* now simplifies to

$$c^* = g^a \; ; \; \pi^* = (g^a)^{x_1 t^* + y_1} \; ; \; K_1^* = h^a . \tag{6}$$

Hence (c^*, π^*) is a correctly generated ciphertext for the real key K_1^* with randomness $a \in \mathbb{Z}_p$ since by Equations (4) and (5) we have $(u^{t^*}v)^a = (g^{x_1t^*+y_1}h^{x_2(t^*-t^*)})^a = (g^a)^{x_1t^*+y_1} = \pi^*$. Note that the experiment does not explicitly know the randomness a, only the values g^a and h^a. Since the changes are purely conceptual we have

$$\Pr[X_3] = \Pr[X_2].$$

Game 4. Consider a query (pred, C) adversary $\mathcal{A}$ makes to the oracle $\mathrm{CD}\textsc{ec}(\cdot,\cdot)$ and recall that $\mathrm{pred} : \mathbb{G} \to \{0,1\}$ is some efficiently computable predicate. After the TCR check, such a query is now processed in the following way. If the ciphertext C is inconsistent (this is checked using ω) it gets rejected. If the ciphertext $C = (c, \pi)$ is consistent (by $C \neq C^*$ at this point we have $t \neq t^*$) compute K as

$$K = \left(\frac{\pi}{c^{x_1t+y_1}}\right)^{\frac{1}{x_2(t-t^*)}} . \tag{7}$$

If $\mathrm{pred}(K) = 0$, then reject, and return K otherwise.

This change is purely conceptual since for any consistent ciphertext with $t \neq t^*$ we have $\pi = (u^tv)^r = (g^{x_1t+y_1}h^{x_2(t-t^*)})^r = c^{x_1t+y_1}K^{x_2(t-t^*)}$ which implies correctness of Equation (7). Consequently,

$$\Pr[X_4] = \Pr[X_3].$$

Game 5. A query (pred, C) adversary $\mathcal{A}$ makes to the oracle $\mathrm{CD}\textsc{ec}(\cdot,\cdot)$ is now processed in the following way. After the TCR check, for all ciphertexts $C = (c, \pi)$ (consistent and inconsistent alike) the key K is decapsulated using Equation (7). If $\mathrm{pred}(K) = 0$, then reject, and return K otherwise.

Note that at this point the experiment does not make use of $\omega = \log_g h$ anymore and hence the value $h \in \mathbb{G}$ from the public key can be generated as a random group element. The proof of the following key lemma will be given later.

Lemma 4.3

$$|\Pr[X_5] - \Pr[X_4]| \leq Q \cdot (\mathrm{uncert}_{\mathcal{A}}(k) + \frac{1}{p})$$

.

Intuitively the lemma holds since for *one* inconsistent ciphertext submitted to the $\mathrm{CD}\textsc{ec}(\cdot,\cdot)$ oracle, the virtual key K computed as in Equation (7) looks like a uniform and undependent element in the view of the adversary (the probability space is the redundancy contained in *sk* that is information-theoretically hidden from *pk*). But for a random independent key K, the probability that $\mathrm{pred}(K) = 1$ (meaning the ciphertext does not get rejected) is bounded by $\mathrm{uncert}_{\mathcal{A}}(k)$ which is negligible by assumption. Hence, with high probability the inconsistent ciphertext gets rejected and the virtual key K remains hidden from the adversary's view. This makes it possible to use a hybrid argument to show that, with high probability, *all* inconsistent ciphertexts get rejected in Game 5, just as in Game 4.

Game 6. The real challenge key K_1^* is replaced by the random key $K_0^* \in \mathbb{G}$. Since in Game 5 we had $K_1^* = h^a$ and apart from that the experiment was run using the values g, h, g^a only (where all three elements are random group elements), we have

$$|\Pr[X_6] - \Pr[X_5]| \leq \mathbf{Adv}^{\mathrm{ddh}}_{\mathbb{G},\mathcal{B}_{\mathrm{ddh}}}(k).$$

(Note that K_1^* is only used for $b = 1$, which occurs with probability 1/2.)

Finally, in Game 6 the distribution of the challenge key K_0^* does not depend on b, and consequently

$$\Pr[X_6] = 1/2\ .$$

Collecting the probabilities proves the theorem. ▮

It leaves to prove Lemma 4.3.

Proof: For $j \in \{1, \ldots, Q\}$, let E_j denote the event that in Game 4, adversary $\mathcal{A}$ submits as j-th decryption query a ciphertext (C_j, pred_j) that gets rejected, but would *not* have been rejected in Game 5. Let $E := E_1 \vee \ldots \vee E_Q$. Analogously, let F denote the event that in Game 5, adversary $\mathcal{A}$ submits at any point a decapsulation query that does not get rejected, but would have been rejected in Game 4. Games 4 and 5 proceed identical unless a decapsulation query gets treated differently. Consequently,

$$\Pr[X_4 \wedge \neg E] = \Pr[X_5 \wedge \neg F] \quad \text{and} \quad \Pr[E_1] + \ldots + \Pr[E_Q] \geq \Pr[E] = \Pr[F]. \tag{8}$$

Now consider events $\hat{E}_j$, where for $j \in \{1, \ldots, Q\}$, event $\hat{E}_j$ denotes that the j-th decryption query (C_j, pred_j) in Game 4 gets rejected, but $\mathrm{pred}_j(K') = 1$ under an independently uniformly chosen symmetric key $K' \overset{\$}{\leftarrow} \mathbb{G}$. By definition of $\mathrm{uncert}(k)$ we have

$$\frac{1}{Q} \cdot \left(\sum_{1 \leq j \leq Q} \Pr[\hat{E}_j] \right) \leq \mathrm{uncert}_{\mathcal{A}}(k),$$

since Game 4 has the same complexity as the original IND-CCCA experiment.[6] We now claim that

$$\text{for all } j\text{:} \quad |\Pr[\hat{E}_j] - \Pr[E_j]| \leq 1/p\ . \tag{9}$$

This implies

$$\Pr[E_1] + \ldots + \Pr[E_Q] \leq \Pr[\hat{E}_1] + \ldots + \Pr[\hat{E}_Q] + \frac{Q}{p} \leq Q \cdot (\mathrm{uncert}_{\mathcal{A}}(k) + \frac{1}{p}).$$

Combining this with (8) and using Lemma 3.2 proves the lemma.

It leaves to prove Equation (9). Fix a security parameter k and $j \in \{1, \ldots, Q(k)\}$. Let $C = (c, \pi)$ be the ciphertext of the j-th decryption query in Game 4.

Let $t := \mathsf{TCR}(c)$, $r := \log_g c$, and $\beta := \log_g \pi$. Write furthermore $\omega = \log_g h$, and $x = \log_g u = x_1 + \omega x_2$, $y = \log_g v = y_1 - t^* \omega x_2$ as before. Then (c, π) is consistent iff $\pi = (u^t v)^r$, or, alternatively, iff $\beta = r \cdot (tx + y)$. Furthermore, if (c, π) is consistent, then E_j and $\hat{E}_j$ cannot be fulfilled by definition. However, we claim that under the condition that (c, π) is inconsistent, the "virtual key" K used to determine whether $\mathrm{pred}(K) = 1$ or not (according to the rules of

[6] There is some "fuzziness" here; depending on the used complexity model, Game 4 might have a complexity which is only roughly that of the IND-CCCA game. Formally, a concrete security analysis requires in that case that adversaries be also "valid when run in slightly more complex environments than the IND-CCCA game." We stress that such an extension to our theory *is* possible, in fact straightforward, and in particular yields the results one would expect. However, such an extension also requires a more complex set of definitions. In the interest of a clear presentation, we stick to the usual but formally non-rigorous convention that slight changes to a security game do not add to its complexity.

Game 5) is, just as the key K' of event $\hat{E}_j$, uniformly distributed and *independent* of the choice of the predicate pred. So assume $\beta \neq r \cdot (tx + y)$. For the key K from Game 5, it holds that

$$\begin{aligned}\log_g K = \log_g \left(\left(\frac{\pi}{c^{tx_1+y_1}} \right)^{\frac{1}{x_2(t-t^*)}} \right) &= \frac{1}{x_2(t-t^*)} \left(\beta - r \cdot (tx_1 + y_1)\right) \\ = \frac{1}{x_2(t-t^*)} \left(\beta - r \cdot (tx + y) - r\omega x_2 \cdot (t - t^*)\right) &= \frac{1}{x_2(t-t^*)} \underbrace{\left(\beta - r \cdot (tx + y)\right)}_{\neq 0} - r\omega\end{aligned}$$

Define the mapping F through

$$F(X) := (X + r\omega) \cdot \frac{t - t^*}{\beta - r \cdot (tx + y)} \mod p.$$

By the assumptions $\beta - r \cdot (tx + y) \neq 0$ and $t - t^* \neq 0$, this mapping is well-defined and bijective and only depends on information known (in an information-theoretic sense) to the adversary. Hence, to show that, from $\mathcal{A}$'s perspective, K is (almost) uniformly distributed, it suffices to show that $F(\log_g K) = 1/x_2 \mod p$ is (almost) uniformly distributed given a public key. But a public key determines only $x = x_1 + \omega x_2$ and $y = y_1 - t^* \omega x_2$. Hence, x_2 (and consequently $1/x_2$) is still uniformly and independently from $\mathcal{A}$'s view distributed over $\{1, \ldots, p - 1\}$. This implies $|\Pr[\hat{E}_j] - \Pr[E_j]| \leq 1/p$ which is Equation (9). ▮

4.3 Comparison with Cramer-Shoup and Kurosawa-Desmedt

The following table summarizes the key-encapsulation part of the (only IND-CPA secure) ElGamal scheme [16], the Cramer-Shoup encryption scheme [14], the Kurosawa-Desmedt scheme [24], and ours.

Scheme	Ciphertext	Encapsulated Key
ElGamal	g^r	h^r
Cramer-Shoup	$g^r, \hat{g}^r, (u^t v)^r$	h^r
Kurosawa-Desmedt	$g^r, \hat{g}^r$	$(u^t v)^r$
Dual Kuroasawa-Desmedt (ours)	$g^r, (u^t v)^r$	h^r

Here $\hat{g}$ is another element from the public-key. Compared to the Cramer-Shoup scheme, the Kurosawa-Desmedt scheme leaves out the value h^r and defines $(u^t v)^r$ out the encapsulated key. Our results shows that it is also possible to leave out the element $\hat{g}^r$ from the ciphertext and that $\pi = (u^t v)^r$ is sufficient to authenticate $c = g^r$. Hence, our scheme can be viewed as the *dual* of (the KEM part of) the Kurosawa-Desmedt scheme. From another point of view, compared to the IND-CPA secure ElGamal scheme our scheme adds one group element $\pi = (u^t v)^r$ to the KEM ciphertext which is sufficient to prove it IND-CCCA secure under the DDH assumption.

From a technical point of view, our scheme mixes Cramer-Shoup like techniques [13] to obtain a form of "plaintext awareness" for inconsistent ciphertexts with an "algebraic trick" from the Boneh-Boyen identity-based encryption scheme [8] to decrypt consistent ciphertexts. Compared to Cramer-Shoup based proofs [12, 14, 24, 2] the most important technical difference, caused by the mentioned ability to decrypt consistent ciphertexts without knowing the full secret key, is that during our simulation the challenge ciphertexts is never made inconsistent. Intuitively this is the reason why we manage to maintain a consistent simulation using less redundancy in the secret key. This demonstrates that IND-CCCA security can be obtained with constructions that inherently differ from hash proof systems.

On the other hand, the security proofs of all schemes based on IBE-techniques [11, 10, 22, 23, 21] inherently relies on some sort of external consistency check for the ciphertexts. This can be seen as the main reason why security of the IBE-based PKE schemes could only be proved in pairing groups (or relative to a gap-assumption), where the pairing was neccessary for helping the proof identifying inconsistent ciphertexts. In our setting, the consistency check is done implicitly, using information-theoretic arguments borrowed from hash proof systems.

4.4 Explicit vs. implicit rejection

The scheme is given in its explicit-rejction variant, i.e. all inconsistent ciphertexts get immediately rejected by the decapsulation algorithm. Following [14] we can also give an implicit-rejection variant of the scheme, where inconsistent ciphertexts yield one uniform key and hence will be rejected by the authenticity property of the DEM. Details are given in Appendix A. The specific structure of the implicit-rejection KEM makes it possible to share the public elements g and h among many parties since decryption does not depend on the knowledge of $\omega = \log_g(h)$ anymore. Hence, similar to the Cramer-Shoup scheme, this implicit-rejection scheme can be used in the setting of multi-recipient encryption [3], where one single message is being simultaneously sent to a set of n different recipients.

4.5 A hash-free variant

Similar to [14] we can also give a hash-free variant of our scheme that abandons the hash function $\mathsf{TCR} : \mathbb{G} \to \mathbb{Z}_p$. This variant is useful when neither a bijective encoding nor a target-colission resistant hash function $\mathcal{TCR}$ is available. In terms of computational efficiency and size of public/secret keys the hash-free variant is slightly less efficient but security can be proved relying solely on the DDH assumption. Details are given in Appendix B.

4.6 Efficiency

We compare our new DDH-based scheme's efficiency with the one of Kurosawa and Desmedt (in its more efficient "explicit-rejection" variant from [29]). Most importantly, the number of exponentiations for encryption and decryption are equal in both schemes. Although our security result is much more general (our KEM can be combined with any authenticated encryption scheme) this is not an exclusive advantage of our scheme. In fact we can derive the same result for the KD scheme from a more general theorem that we will prove in Section 6. (A similar result about combining the Kurosawa-Desmedt scheme with authenticated encryption was already obtained in [4] in the context of statefull encryption.)

However, there is one crucial difference in case one needs a scheme that is provably secure *solely* on the DDH assumption. Note that security (of the KD scheme and ours) relies on the DDH assumption *and* the assumption that $\mathcal{TCR}$ is target collision resistant. So as long as one does not want to sacrifice *provable* security by implementing the TCR function with a dedicated hash function like SHA-x or MD5 (what potentially renders the whole scheme insecure given the recent progress in attacking certain hash functions [38, 39]), one must either resort to inefficient generic constructions of TCR functions [25, 33], or one can use the "hash-free technique" described in [14]. With this latter technique, one can get rid of the TCR function completely; however, this comes at the cost of additional elements in the public and the secret key, and additional exponentiations during encryption. This overhead is linear in the number of elements that would have been hashed with the TCR. In the Kurosawa-Desmedt scheme, TCR acts on two group elements whereas in our scheme only on one. Hence the hash-free variant of our scheme is more efficient.

More importantly, since in our scheme a TCR is employed which maps *one* group element to integers modulo the group-order this can also be a bijection. In many concrete groups, e.g., when using the subgroup of quadratic residues modulo a safe prime or certain elliptic curves, this bijection can be trivially implemented at zero cost [14, 10], without any additional computational assumption, and without sacrificing provable security. See Appendix C for more details. In terms of efficiency we view this as the main benefit of our scheme.

5 Key Encapsulation from n-Linear

5.1 Linear Assumptions

Let $n = n(k)$ be a polynomial in k. Generalizing [9, 21] we introduce the class of n-Linear assumptions which can be seen as a natural generalization of the DDH assumption and the Linear assumption.

Let $\mathcal{GS}$ be a group scheme. We define the n-lin-advantage of an adversary $\mathcal{B}_{n\text{-lin}}$ as

$$\begin{aligned}\mathbf{Adv}^{n\text{-lin}}_{\mathcal{GS},\mathcal{B}_{n\text{-lin}}}(k) = \frac{1}{2}\Big| \Pr[\mathcal{B}_{n\text{-lin}}(g_1,\ldots,g_n,g_1^{r_1},\ldots,g_n^{r_n},h,h^{r_1+\ldots+r_n}) = 1] \\ - \Pr[\mathcal{B}_{n\text{-lin}}(g_1,\ldots,g_n,g_1^{r_1},\ldots,g_n^{r_n},h,K) = 1]\Big|,\end{aligned}$$

where $g_1,\ldots,g_n,h,K \stackrel{\$}{\leftarrow} \mathbb{G}$ and all $r_i \leftarrow \mathbb{Z}_p^*$. We say that the *$n$-Linear Decisional Diffie-Hellman (n-Linear) assumption relative to group scheme* $\mathcal{GS}$ holds if $\mathbf{Adv}^{n\text{-lin}}_{\mathcal{GS},\mathcal{B}_{n\text{-lin}}}$ is a negligible function in k for all polynomial-time adversaries $\mathcal{B}_{n\text{-lin}}$.

The n-Linear assumptions form a strict hierarchy of security assumptions with 1-Linear = DDH, 2-Linear=Linear [9] and, the larger the n, the weaker the n-Linear assumption. More precisely, for any $n \geq 1$ we have that n-Linear implies n+1-Linear. On the other hand (extending the case of $n = 1$ [9]) we can show that in the generic group model [35], the n+1-Linear assumption holds, even relative to an n-Linear oracle.

Lemma 5.1 DDH = 1-Linear $\overset{\not\Leftarrow}{\Rightarrow}$ 2-Linear $\overset{\not\Leftarrow}{\Rightarrow}$ 3-Linear $\overset{\not\Leftarrow}{\Rightarrow}$ ...

5.2 The key-encapsulation mechanism

Let $\mathcal{GS}$ be a group scheme where $\mathcal{GR}_k$ specifies $(\hat{\mathbb{G}}, \mathbb{G}, g, p)$ and let $\mathsf{TCR} : \mathbb{G}^{n+1} \to \mathbb{Z}_p$ be a target collision resistant hash function. Generalizing the Kurosawa-Desmedt KEM, for a parameter $n = n(k) \geq 1$, we build $\mathcal{KEM} = (\mathsf{KEM.Kg}, \mathsf{KEM.Enc}, \mathsf{KEM.Dec})$ as follows.

Key generation $\mathsf{KEM.Kg}(1^k)$ generates random group elements $g_1,\ldots,g_n,h \in \mathbb{G}$. Furthermore, it defines $u_j = g_j^{x_j} h^z$ and $v_j = g_j^{y_j} h^{z'}$ for random $z, z' \in \mathbb{Z}_p$ and $x_j, y_j \in \mathbb{Z}_p$ ($j \in \{1,\ldots,n\}$). The public key contains the elements h, $(g_j, u_j)_{1 \leq i \leq n}$, and the secret key contains all corresponding indices.

```
$\mathsf{KEM.Enc}(pk)$
    $\forall j \in \{1,\ldots,n\}$: $r_j \stackrel{\$}{\leftarrow} \mathbb{Z}_p^*$ ; $c_j \leftarrow g_j^{r_j}$
    $d \leftarrow h^{r_1+\ldots+r_n}$ ; $t \leftarrow \mathsf{TCR}(c_1,\ldots,c_n,d)$
    $C \leftarrow (c_1,\ldots,c_n,d)$ ; $K = \prod_{i=1}^n (u_i^t v_i)^{r_i}$
    Return $(C, K)$
```

```
$\mathsf{KEM.Dec}(sk, C)$
    $\forall j \in \{1,\ldots,n\}$: check if $c_j \in \mathbb{G}$
    Check if $d \in \mathbb{G}$
    $t \leftarrow \mathsf{TCR}(c_1,\ldots,c_n,d)$
    Return $K \leftarrow d^{zt+z'} \cdot \prod_{j=1}^n c_j^{x_j t + y_j}$
```

Ciphertexts contain $n+1$ group elements, public/secret keys $2n+1$ elements. The scheme instantiated with $n=1$ precisely reproduces the KEM part of the Kurosawa-Desmedt encryption scheme [24]. We remark that the scheme is presented in its implicit-rejection variant. It is also possible to give a variant that explicitly rejects inconsistent ciphertexts.

Security of the schemes can be explained using the more general framework of computational hash-proof systems. This will be done in Section 6.

Theorem 5.2 Let $\mathcal{GS}$ be a group scheme where the n-Linear problem is hard, assume $\mathcal{TCR}$ is target collision resistant. Then $\mathcal{KEM}$ is secure in the sense of IND-CCCA.

5.3 The Dual Key Encapsulation Mechanism

Let $\mathcal{GS}$ be a group scheme where $\mathcal{GR}_k$ specifies $(\hat{\mathbb{G}}, \mathbb{G}, g, p)$ and let $\mathsf{TCR} : \mathbb{G}^n \to \mathbb{Z}_p$ be a target collision resistant hash function. Generalizing our scheme from Section 4 for a parameter $n = n(k) \geq 1$, we build $\mathcal{KEM} = (\mathsf{KEM.Kg}, \mathsf{KEM.Enc}, \mathsf{KEM.Dec})$ as follows.

Key generation $\mathsf{KEM.Kg}(1^k)$ generates random group elements $g_1, \ldots, g_n, h \in \mathbb{G}$ and indices ω_j such that $h = g_j^{\omega_j}$. Furthermore it defines $u_j = g_j^{x_j}$ and $v_j = g_j^{y_j}$ for random $x_j, y_j \in \mathbb{Z}_p$ ($j \in \{1, \ldots, n\}$). The public key contains the elements h, $(g_j, u_j, v_j)_{1 \leq i \leq n}$, and the secret key contains all corresponding indices.

```
KEM.Enc($pk$)
  $\forall j \in \{1,\ldots,n\}$: $r_j \xleftarrow{\$} \mathbb{Z}_p^*$ ; $c_j \leftarrow g_j^{r_j}$
  $t \leftarrow \mathsf{TCR}(c_1,\ldots,c_n)$
  $\pi \leftarrow \prod_{j=1}^n (u_j^t v_j)^{r_j}$
  $C \leftarrow (c_1,\ldots,c_n,\pi)$ ; $K \leftarrow h^{r_1+\cdots+r_n}$
  Return $(C,K)$
```

```
KEM.Dec($sk, C$)
  $\forall j \in \{1,\ldots,n\}$: check if $c_j \in \mathbb{G}$
  $t \leftarrow \mathsf{TCR}(c_1,\ldots,c_n)$
  If $\prod_{j=1}^n c_j^{x_j t + y_j} \neq \pi$ return $\perp$
  Return $K \leftarrow \prod_{j=1}^n c_j^{\omega_j}$
```

Correctness of the scheme can be verified analogously to Section 4. Ciphertexts contain $n+1$ group elements, public/secret keys $3n+1$ elements. The scheme instantiated with $n=1$ precisely reproduces our DDH-based dual KD-KEM from Section 4.

Theorem 5.3 Let $\mathcal{GS}$ be a group scheme where the n-Linear problem is hard, assume $\mathcal{TCR}$ is target collision resistant. Then $\mathcal{KEM}$ is secure in the sense of IND-CCCA. In particular,

$$\mathbf{Adv}^{\mathrm{ccca}}_{\mathcal{KEM},t,Q,\mathrm{uncert}(k),\mathcal{E}}(k) \leq \mathbf{Adv}^{n\text{-lin}}_{\mathcal{GS},t}(k) + \mathbf{Adv}^{\mathrm{tcr}}_{\mathcal{TCR},t}(k) + \mathrm{uncert}(k) + \frac{Qn}{p} \ .$$

for a suitable environment $\mathcal{E}$ that roughly has the same complexity as the IND-CCCA experiment.

The proof of Theorem 5.3 is similar to the one of Theorem 4.1. We quickly sketch the simulation of the adversary's view. Given the values $(g_1, \ldots, g_n, c_1^* = g_1^{r_1}, \ldots, c_n^* = g_n^{r_n}, h, T)$ from the n-Linear problem (where $T = h^{r_1+\cdots+r_n}$ or random) the simulator picks random $\tilde{x_1}, \ldots, \tilde{x_n}, \tilde{y_1}, \ldots, \tilde{y_n}, z \in \mathbb{Z}_p^*$ and defines the values $(u_i)_{1 \leq i \leq n}$ and $(v_i)_{1 \leq i \leq n}$ as

$$u_i = g_i^{\tilde{x_i}} \cdot h^z, \quad v_i = g_i^{\tilde{y_i}} \cdot h^{-t^* z},$$

where $t^* = \mathsf{TCR}(c_1^*, \ldots, c_n^*)$. Note that (in an information theoretic sense) through the public-key the adversary knows exactly $2n$ linear equations in the $2n+1$ variables $\tilde{x_1}, \ldots, \tilde{x_n}, \tilde{y_i}, \ldots, \tilde{y_n}, z$. The challenge ciphertext is defined as $C^* = (c_1^*, \ldots, c_n^*, \prod_{i=1}^n c_i^{*\tilde{x_i} t^* + \tilde{y_i}})$ and the encapsulated key as T which is either the real key $h^{r_1+\dots+r_n}$ or a random key. For a CCCA decapsulation query (C, pred) the virtual key K is computed as

$$K = \left(\frac{\pi}{\prod_{i=1}^{n} c_1^{\tilde{x_i} t + \tilde{y_i}}} \right)^{\frac{1}{z(t-t^*)}}$$

and K is only returned to the adversary if $\mathrm{pred}(K) = 1$. As in the proof of Theorem 4.1 it can be verified that (i) if C is consistent then K is the correct key; (ii) if C is inconsistent then (from the adversary's view) K is a uniform random element in $\mathbb{G}$ and hence, with high probability, the query will be rejected by the predicate test.

6 Key encapsulation from Hash Proof Systems

In [13] Cramer and Shoup showed that their original scheme in [14] was a special instance of a generic framework based on hash proof systems (HPS). In this section we further elaborate on the usefulness of Constrained chosen-ciphertext secure KEMs by showing that they can be built from any hash-proof system.

Following [24] we recall the basic ideas and show how to build IND-CCCA secure key encapsulation based on a computational variant of hash proof systems. Here we use a slightly different notation that reflects our primary use of hash-proof systems as key-encapsulation mechanisms.

6.1 Hash proof systems

Let $\mathcal{C}, \mathcal{K}$ be sets and $\mathcal{V} \subset \mathcal{C}$ a language. Let $\mathsf{D}_{sk} : \mathcal{C} \to \mathcal{K}$ be a hash function indexed with $sk \in \mathcal{S}$, where $\mathcal{S}$ is a set. A hash function D_{sk} is *projective* if there exists a projection $\mu : \mathcal{S} \to \mathcal{P}$ such that $\mu(sk) \in \mathcal{P}$ defines the action of D_{sk} over the subset $\mathcal{V}$. That is, for every $C \in \mathcal{V}$, the value $K = \mathsf{D}_{sk}(C)$ is uniquely determined by $\mu(sk)$ and C. In contrast, nothing is guaranteed for $C \in \mathcal{C} \setminus \mathcal{V}$, and it may not be possible to compute $\mathsf{D}_{sk}(C)$ from $\mu(sk)$ and C. A *strongly universal*$_2$ projective hash function has the additional property that for $C \in \mathcal{C} \setminus \mathcal{V}$, the projection key $\mu(sk)$ actually says nothing about the value of $K = \mathsf{D}_{sk}(C)$, even given an instance (C^*, K^*) such that $C^* \in \mathcal{C} \setminus \mathcal{V}$ and $K^* = \mathsf{D}_{sk}(C)$. More precisely, for all $pk \in \mathcal{P}$, C, all $C^* \in \mathcal{C} \setminus \mathcal{V}$ with $C \neq C^*$, all $K, K^* \in \mathcal{K}$,

$$\Pr_{\substack{sk \in \mathcal{S} \\ \mathsf{D}_{sk}(C^*) = K^* \\ \mu(sk) = pk}} [\mathsf{D}_{sk}(C) = K] = 1/|\mathcal{K}|. \tag{10}$$

A hash proof system $\mathcal{HPS} = (\mathsf{HPS.param}, \mathsf{HPS.pub}, \mathsf{HPS.priv})$ consists of three algorithms. The randomized algorithm $\mathsf{HPS.param}(1^k)$ generates instances of $params = (group, \mathcal{C}, \mathcal{V}, \mathcal{P}, \mathcal{S}, \mathsf{D}_{(\cdot)} : \mathcal{C} \to \mathcal{K}, \mu : \mathcal{S} \to \mathcal{P})$, where *group* may contain some additional structural parameters. The deterministic public evaluation algorithm $\mathsf{HPS.pub}$ inputs the projection key $pk = \mu(sk)$, $C \in \mathcal{V}$ and a witness w of the fact that $C \in \mathcal{V}$ and returns $K = \mathsf{D}_{sk}(C)$. The deterministic private evaluation algorithm inputs $sk \in \mathcal{S}$ and returns $\mathsf{D}_{sk}(C)$, without knowing a witness. We further assume there are efficient algorithms given for sampling $sk \in \mathcal{S}$ and sampling $C \in \mathcal{V}$ uniformly together with a witness w.

As computational problem we require that the *subset membership problem* is hard in $\mathcal{HPS}$ which means that the two elements C and C' are computationally indistinguishable, for random $C \in \mathcal{V}$ and random $C' \in \mathcal{C} \setminus \mathcal{V}$. This is captured by defining the advantage function $\mathbf{Adv}^{\mathrm{sm}}_{\mathcal{HPS},\mathcal{A}}(k)$

of an adversary $\mathcal{A}$ as

$$\begin{aligned}\mathbf{Adv}^{\mathrm{sm}}_{\mathcal{HPS},\mathcal{A}}(k) \;:=\; \big| \Pr[C_1 \xleftarrow{\$} \mathcal{C}\,;\, b' \xleftarrow{\$} \mathcal{A}(\mathcal{C},\mathcal{V},C_1) \;:\; b' = 1\,] \\ -\, \Pr[C_0 \xleftarrow{\$} \mathcal{C}\setminus\mathcal{V}\,;\, b' \xleftarrow{\$} \mathcal{A}(\mathcal{C},\mathcal{V},C_0) : \; b' = 1\,]\big| \,.\end{aligned}$$

6.2 Key encapsulation from HPS

Using the above notion of a hash proof system, Kurosawa and Desmedt [24] proposed a hybrid encryption scheme which improved the schemes from [13]. The key-encapsulation part of it is as follows. The system parameters of the scheme consist of $params \xleftarrow{\$} \mathsf{HPS.param}(1^k)$.

$\mathsf{KEM.Kg}(k)$**.** Choose random $sk \xleftarrow{\$} \mathcal{S}$ and define $pk = \mu(sk) \in \mathcal{P}$. Return (pk, sk).

$\mathsf{KEM.Enc}(pk)$**.** Pick $C \xleftarrow{\$} \mathcal{V}$ together with its witness ω that $C \in \mathcal{V}$. The session key $K = \mathsf{D}_{sk}(C) \in \mathcal{K}$ is computed as $K \xleftarrow{\$} \mathsf{HPS.pub}(pk, C, \omega)$. Return (K, C).

$\mathsf{KEM.Dec}(sk, C)$**.** Reconstruct the key $K = \mathsf{D}_{sk}(C)$ as $K \leftarrow \mathsf{HPS.priv}(sk, C)$ and return K.

We can prove the following theorem that is a slight generalization of [24].

Theorem 6.1 If $\mathcal{HPS}$ is strongly universal$_2$ and the subset membership problem is hard in $\mathcal{HPS}$ then $\mathcal{KEM}$ is secure in the sense of IND-CCCA.

Unfortunately, the original KEM part of the Kurosawa Desmedt DDH-based hybrid encryption scheme [24] cannot be explained using this framework and hence needed a separate proof of security. This is since the underlying DDH-based hash proof system involves a target collision resistant hash function TCR which is a "computational primitive" whereas the strongly universal$_2$ property from Equation (10) is a *statistical property* which is in particular not fulfilled by the DDH-based HPS from [13] used in [24]. In fact, the most efficient HPS-based schemes that are known involve computation of a TCR function and hence all need a separate proof of security. We note that this problem is inherited from the original HPS approach [14].

We overcome this problem we defining the weaker notion of *computational hash proof systems.*

6.3 Computational hash proof systems

We now define a weaker computational variant of strongly universal$_2$ hashing. To an adversary $\mathcal{B}$ we associate the following experiment $\mathbf{Exp}^{\mathrm{cu}_2}_{\mathcal{HPS},\mathcal{B}}$.

```
Experiment $\mathbf{Exp}^{\mathrm{cu}_2}_{\mathcal{HPS},\mathcal{B}}(k)$
    $params \xleftarrow{\$} \mathsf{HPS.param}(1^k)\,;\; sk \xleftarrow{\$} \mathcal{S}\,;\; pk \leftarrow \mu(sk)$
    $C^* \xleftarrow{\$} \mathcal{C}\setminus\mathcal{V}\,;\; K^* \leftarrow \mathsf{D}_{sk}(C^*)\,;\; (C, St) \xleftarrow{\$} \mathcal{B}_1^{\textsc{EvalD}(\cdot)}(pk, C^*, K^*)$
    $b \xleftarrow{\$} \{0,1\}\,;\; K_0 \xleftarrow{\$} \mathcal{K}\,;\; K_1 \leftarrow \mathsf{D}_{sk}(C)$
    $b' \xleftarrow{\$} \mathcal{B}_2(St, K_b)$
    If $b = b'$ return 1 else return 0
```

where the evaluation oracle $\textsc{EvalD}(C)$ returns $K = \mathsf{D}_{sk}(C)$ if $C \in \mathcal{V}$ and $\perp$, otherwise. We also restrict to adversaries that only return ciphertexts $C \neq C^*$ and that ensure $C \in \mathcal{C}\setminus\mathcal{V}$. This is without losing generality, since $\mathcal{B}_1$ can check $C \in \mathcal{V}$ with its oracle EVALD. We define the advantage of $\mathcal{B}$ in the experiment as

$$\mathbf{Adv}^{\mathrm{cu}_2}_{\mathcal{HPS},\mathcal{B}}(k) \;=\; \left|\Pr[\mathbf{Exp}^{\mathrm{cu}_2}_{\mathcal{HPS},\mathcal{B}}(k) = 1] - \frac{1}{2}\right| .$$

A hash proof system $\mathcal{HPS}$ is said to be *computationally universal*$_2$ (CU$_2$) if for all polynomial-time adversaries $\mathcal{B}$ that satisfy these requirements, the advantage function $\mathbf{Adv}^{\mathrm{cu}_2}_{\mathcal{HPS},\mathcal{B}}(k)$ is a negligible function in k.

The following theorem strengthens Theorem 6.1.

Theorem 6.2 If $\mathcal{HPS}$ is computationally universal$_2$ and the subset membership problem is hard then $\mathcal{KEM}$ from Section 6.2 is IND-CCCA secure. In particular,

$$\mathbf{Adv}^{\mathrm{ccca}}_{\mathcal{KEM},t,Q,\mathrm{uncert}(k)}(k) \le \mathbf{Adv}^{\mathrm{sm}}_{\mathcal{HPS},t}(k) + Q \cdot \mathrm{uncert}(k) + (2Q+1) \cdot \mathbf{Adv}^{\mathrm{cu}_2}_{\mathcal{HPS},t}(k)).$$

Proof: Let $\mathcal{A}$ be an adversary on the IND-CCCA security of the KEM. We will consider a sequence of games, Game 1, Game 2, ..., each game involving $\mathcal{A}$. Let X_i be the event that in Game i, it holds that $b = b'$, i.e., that the adversary succeeds.

Game 1. The KEM IND-CCCA game with random $b \in \{0, 1\}$, i.e., we have

$$|\Pr[X_1] - 1/2| = \mathbf{Adv}^{\mathrm{ccca}}_{\mathcal{KEM},\mathcal{A}}(k) .$$

As this point we can assume that the real key K_1^* is computed as $K_1^* = \mathsf{D}_{sk}(C^*)$ for a uniformly chosen $C^* \xleftarrow{\$} \mathcal{V}$ whereas the random key K_0^* is computed as $K_0^* \xleftarrow{\$} \mathcal{K}$.

Game 2. Replace the challenge ciphertext with $C^* \xleftarrow{\$} \mathcal{C} \setminus \mathcal{V}$ and still create the real challenge key as $K_1^* = \mathsf{D}_{sk}(C^*)$.

$$|\Pr[X_2] - \Pr[X_1]| \le \mathbf{Adv}^{\mathrm{sm}}_{\mathcal{HPS},\mathcal{A}}(k)$$

Game 3. Let (pred_j, C_j) be the jth decapsulation query made by $\mathcal{A}$. Decapsulation now checks if $C \in \mathcal{V}$. (Note that this check needs not be efficiently implementable.) If yes it returns $K = \mathsf{D}_{sk}(C)$. If not, it rejects.

Let E_j be the event that $C_j \in \mathcal{C} \setminus \mathcal{V}$ but $\mathrm{pred}_j(K_j) = 1$, and define $\overrightarrow{E}_j = E_1 \vee \ldots \vee E_j$, and $E = \overrightarrow{E}_Q$. Since unless E happens, the Games 2 and 3 proceed indentically, we have

$$|\Pr[X_3] - \Pr[X_2]| \le \Pr[E].$$

We now upper bound $\Pr[E]$. Let $\hat{E}_j$ be the event that $C_j \in \mathcal{C} \setminus \mathcal{V}$ but $\mathrm{pred}_j(K') = 1$ for an independently uniformly chosen key $K' \in \mathcal{K}$.

To do so, we define the following adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ against CU$_2$ of $\mathcal{HPS}$. Adversary $\mathcal{B}_1$ inputs (pk, C_1^*, K_1^*) and first uniformly chooses $j^* \in \{1, \ldots, Q\}$. It then provides $\mathcal{A}$ with (pk, C_1^*, K_1^*). For $1 \le i \le j^* - 1$, adversary $\mathcal{A}$'s ith decapsulation query (pred_i, C_i) is processed as follows. First $\mathcal{B}_1$ calls EVALD(C_i) to obtain K_i or $\perp$. If the answer was $\perp$ (which means that $C_i \notin \mathcal{V}$) it returns $\perp$ to $\mathcal{A}$. Otherwise ($C_i \in \mathcal{V}$) $\mathcal{B}_1$ defines $K_i \leftarrow$ EVALD$(C_i) = \mathsf{KEM.Dec}(sk, C_i)$ and returns K_i if $\mathrm{pred}_i(K_i) = 1$ and $\perp$, otherwise.

Adversary $\mathcal{A}$'s j^*-th decapsulation query $(\mathrm{pred}_{j^*}, C_{j^*})$ is processed as follows. If EVALD(C_{j^*}) yields $\perp$ (meaning $C_{j^*} \in \mathcal{V}$) then $\mathcal{B}_1$ terminates and lets $\mathcal{B}_2$ return $b' = 0$. Otherwise, $\mathcal{B}_1$

returns C_{j^*} to its own CU_2 experiment. Finally, $\mathcal{B}_2$ inputs a challenge key K_b and returns $b' = \mathrm{pred}_{j^*}(K_b)$ to its experiment and terminates.

We now analyze $\mathcal{B}$'s success probability. In analogy to the events $\hat{E}_j$, define events F_j and $\hat{F}_j$. Here, F_j denotes the probability that *in the setting which $\mathcal{B}$ simulates for $\mathcal{A}$* (i.e., in a setting in which all $\mathcal{A}$ gets all queries $C \in \mathcal{C} \setminus \mathcal{V}$ rejected), the j-th query C_j is $\in \mathcal{C} \setminus \mathcal{V}$ but $\mathrm{pred}_j(K_j) = 1$. Analogously, $\hat{F}_j$ denotes the event that in this setting, C_j is $\in \mathcal{C} \setminus \mathcal{V}$ but $\mathrm{pred}_j(K'_j) = 1$ for a uniformly and independently chosen key K'_j. Define $\overrightarrow{F}_j = F_1 \vee \ldots \vee F_j$ and $\overrightarrow{\hat{F}}_j = \hat{F}_1 \vee \ldots \vee \hat{F}_j$ as above.

Now we inductively show that for all j, we have $\Pr[\overrightarrow{F}_j] = \Pr[\overrightarrow{E}_j]$ and $\Pr[\overrightarrow{F}_j] = \Pr[\overrightarrow{E}_j]$. For $j = 1$, this is clear from the definition. For $j > 1$, we have inductively

$$\begin{aligned}\Pr[\overrightarrow{F}_j] &= \Pr[\overrightarrow{F}_j \wedge \overrightarrow{F}_{j-1}] + \Pr[\overrightarrow{F}_j \wedge \neg\overrightarrow{F}_{j-1}] = \Pr[\overrightarrow{F}_{j-1}] + \Pr[\overrightarrow{F}_j \mid \neg\overrightarrow{F}_{j-1}] \cdot \Pr[\neg\overrightarrow{F}_{j-1}]\\ &\overset{(*)}{=} \Pr[\overrightarrow{E}_{j-1}] + \Pr[\overrightarrow{E}_j \mid \neg\overrightarrow{E}_{j-1}] \cdot \Pr[\neg\overrightarrow{E}_{j-1}]\\ &= \Pr[\overrightarrow{E}_{j-1}] + \Pr[\overrightarrow{E}_j \mid \neg\overrightarrow{E}_{j-1}] \cdot \Pr[\neg\overrightarrow{E}_{j-1}] = \Pr[\overrightarrow{E}_j].\end{aligned}$$

In $(*)$, we use not only the induction hypothesis, but also the fact that $Pr[\overrightarrow{F}_j \mid \neg\overrightarrow{F}_{j-1}] = Pr[\overrightarrow{E}_j \mid \neg\overrightarrow{E}_{j-1}]$ for every j by definition of E_j and F_j. Similarly, we can show $\Pr[\overrightarrow{\hat{F}}_j] = \Pr[\overrightarrow{\hat{E}}_j]$ for arbitrary $j \in \{1, \ldots, Q\}$.

We split $\mathbf{Adv}^{\mathrm{cu}_2}_{\mathcal{HPS},t}(k)$ into

$$\begin{aligned}\mathbf{Adv}^{\mathrm{cu}_2}_{\mathcal{HPS},t}(k) &= \left|\frac{1}{2} \cdot \Pr[b' = 1 \mid b = 1] + \frac{1}{2} \cdot \Pr[b' = 0 \mid b = 0] - \frac{1}{2}\right|\\ &= \frac{1}{2} \cdot \big|\Pr[b' = 1 \mid b = 1] - \Pr[b' = 1 \mid b = 0]\big|\end{aligned}$$

and consider the two summands seperately. Now note that by definition of $\mathcal{B}$, we have

$$\begin{aligned}\Pr[b' = 1 \mid b = 1] &= \frac{1}{Q}\sum_{j=1}^{Q} \Pr[F_j] \geq \frac{1}{Q}\Pr[F]\\ &= \frac{1}{Q}\sum_{j=1}^{Q} \Pr[F_j \mid \overrightarrow{F}_{j-1}] = \frac{1}{Q}\sum_{j=1}^{Q} \Pr[E_j \mid \overrightarrow{E}_{j-1}] = \frac{1}{Q}\Pr[E],\end{aligned}$$

whereas

$$\Pr[b' = 1 \mid b = 0] = \frac{1}{Q}\sum_{j=1}^{Q} \Pr[\hat{F}_j] = \mathrm{uncert}_{\mathcal{A}}(k),$$

since Game 2 has the same complexity as the original IND-CCCA game.[7] Summarizing, we obtain

$$|\Pr[X_3] - \Pr[X_2]| \leq \Pr[E] \leq Q \cdot \left(2\mathbf{Adv}^{\mathrm{cu}_2}_{\mathcal{HPS},t}(k) + \mathrm{uncert}_{\mathcal{A}}(k)\right),$$

[7] cf. footnote 6

Game 4. The real challenge key K_1^* is replaced by the random key $K_0^* \in \mathcal{K}$. We have

$$|\Pr[X_4] - \Pr[X_3]| \leq \mathbf{Adv}^{\mathrm{cu}_2}_{\mathcal{HPS},t}(k)$$

Finally, since in Game 4 the distribution of the challenge key K_0^* is independent of b we have

$$\Pr[X_4] = 1/2 \ .$$

Collecting the probabilities proves the theorem. ▌

6.4 A computational HPS from n-Linear

Let $\mathcal{GS}$ be a group scheme where $\mathcal{GR}_k$ specifies $(\hat{\mathbb{G}}, \mathbb{G}, g, p)$. Let *group* $= (\mathcal{GR}, g_1, \ldots, g_n, h)$, where $g_1, \ldots, g_n, h$ are independent generators of $\mathbb{G}$. Define $\mathcal{C} = \mathbb{G}^{n+1}$ and $\mathcal{V} = \{(g_1^{r_1}, \ldots, g_n^{r_n}, h^{r_1+\ldots+r_n}) \subset \mathbb{G}^{n+1} \ : \ r_1, \ldots, r_n \in \mathbb{Z}_p\}$ The values $(r_1, \ldots, r_n) \in \mathbb{Z}_p^n$ are a witness of $C \in \mathcal{V}$. Let $\mathsf{TCR} : \mathbb{G}^{n+1} \to \mathbb{Z}_p$ be a target collision resistant hash function. Let $\mathcal{S} = \mathbb{Z}_p^{2n+2}$, $\mathcal{P} = \mathbb{G}^{2n}$, and $\mathcal{K} = \mathbb{G}$. For $sk = (x_1, y_1, \ldots, x_n, y_n, z, z') \in \mathbb{Z}^{2n+2}$, define $\mu(sk) = (u_1, \ldots, u_n, v_1, \ldots, v_n)$, where, for $1 \leq i \leq n$, $u_i = g_i^{x_i} h^z$ and $v_i = g_i^{y_i} h^{z'}$. This defines the output of $\mathsf{HPS.param}(1^k)$. For $C = (c_1, \ldots, c_n, d) \in \mathcal{C}$ define

$$\mathsf{D}_{sk}(C) := d^{zt+z'} \cdot \prod_{i=1}^{n} c_i^{x_i t + y_i}, \ \text{where } t = \mathsf{TCR}(c_1, \ldots, c_n) \ . \tag{11}$$

This defines $\mathsf{HPS.priv}(sk, C)$. Given $pk = \mu(sk)$, $C \in \mathcal{V}$ and a witness $w = (r_1, \ldots, r_n) \in (\mathbb{Z}_p)^n$ such that $C = (c_1, \ldots, c_n, d) = (g_1^{r_1}, \ldots, g_n^{r_n}, h^{r_1+\ldots+r_n})$ public evaluation $\mathsf{HPS.pub}(pk, C, w)$ computes $K = \mathsf{D}_{sk}(C)$ as

$$K = \prod_{i=1}^{n} (u_i^t v_i)^{r_i} \ .$$

Correctness follows by Equation (11) and the definition of μ. This completes the description of $\mathcal{HPS}$. Clearly, under the n-Linear assumption, the subset membership problem is hard in $\mathcal{HPS}$.

Obviously, the above defined HPS is not strongly universal$_2$ in the sense of Equation (10). But it is still computationally universal$_2$.

Lemma 6.3 The n-Linear based HPS is computationally universal$_2$.

Together with Theorem 6.2 this proves Theorem 5.2. For the case $n = 1$ this also gives an alternative security proof for the Kurosawa-Desmedt scheme [24].

Proof: Consider an adversary $\mathcal{B}$ in the CU$_2$ experiment such that $\mathcal{B}_1$ outputs a ciphertext $C \in \mathcal{C} \setminus \mathcal{V}$ and let $K \leftarrow \mathsf{D}_{sk}(C)$. Let COL be the event that $C \neq C^*$ but $\mathsf{TCR}(C) = \mathsf{TCR}(C^*)$. We claim that for the following adversary $\mathcal{B}_{\mathrm{tcr}}$ we have $\mathbf{Adv}^{\mathrm{tcr}}_{\mathsf{TCR},\mathcal{B}_{\mathrm{tcr}}}(k) = \Pr[\mathrm{COL}]$. Adversary $\mathcal{B}_{\mathrm{tcr}}$ inputs (s, C^*) and generates a random instance of *params* with known indices α_i such that $h = g^{\alpha_i}$. Furthermore, $\mathcal{B}_{\mathrm{tcr}}$ picks a random $sk \in \mathcal{S}$ and runs $\mathcal{B}_1$ on $pk = \mu(sk)$, a random $C^* \in \mathcal{C} \setminus \mathcal{V}$, and $K^* = \mathsf{D}_{sk}(C^*)$. To answer a query to the evaluation oracle $\textsc{Eval}\mathsf{D}(\cdot)$, $\mathcal{B}_{\mathrm{tcr}}$ fist verifies $C = (c_1, \ldots, c_n, d) \in \mathcal{V}$ by checking if $\prod c_i^{\alpha_i} = d$. If not, return $\perp$. Otherwise it returns $K = \mathsf{D}_{sk}(C)$. If for a decapsulation query C event COL happens, $\mathcal{B}_{\mathrm{tcr}}$ returns C to its TCR experiment and terminates.

Now we claim that conditioned under $\neg$COL, the key $K = \mathsf{D}_{sk}(C)$ is a uniform element in $\mathcal{K}$ independent of the adversary's view. This implies that not even a *computationally unbounded*

$\mathcal{B}_2$ could succeed in the second stage. Hence, $\mathbf{Adv}^{\text{cu}_2}_{\mathcal{HPS},\mathcal{B}}(k) \leq \mathbf{Adv}^{\text{tcr}}_{\mathsf{TCR},\mathcal{B}_{\text{tcr}}}(k)$, which proves the lemma.

Let $\log(\cdot) = \log_g(\cdot)$. Consider the view of $\mathcal{B}_2$ consisting of the random variables (pk, C^*, K^*, C), where $sk = (x_1, y_1, \ldots, x_n, y_n, z, z') \overset{\$}{\leftarrow} \mathbb{Z}^{2n+2}$, $pk = \mu(sk) = (u_1, \ldots, u_n, v_1, \ldots, v_n)$, $C^* = (c_1^*, \ldots, c_n^*, d^*) = (g_1^{r_1^*}, \ldots, g_n^{r_n^*}, h^{r^*})$ with $\sum r_i^* \neq r^*$ since $C^* \in \mathcal{C} \setminus \mathcal{V}$, $K^* = \mathsf{D}_{sk}(C^*)$, and $C = (c_1, \ldots, c_n, d) = (g_1^{r_1}, \ldots, g_n^{r_n}, h^r)$ ($\sum r_i \neq r$ since $C \in \mathcal{C} \setminus \mathcal{V}$). From the system parameters $g_1, \ldots, g_n, h$, adversary $\mathcal{B}_2$ learns $\omega = \log h$, $\omega_i = \log g_i$, and from pk

$$\text{for } 1 \leq i \leq n \quad : \quad \log u_i = \omega_i x_i + \omega z, \quad \log v_i = \omega_i y_i + \omega z' \; . \tag{12}$$

From C^* the adversary learns $r_i^* = \log_{g_i} c_i^*$, $r^* = \log_h d^*$, and from K^* (by Equation (11)) the value

$$\log K^* = \sum \omega_i r_i^* (x_i t^* + y_i) + \omega(z t^* + z') \; , \tag{13}$$

and $t^* = \mathsf{TCR}(c_1^*, \ldots, c_n^*, d^*)$. Furthermore, from C, $\mathcal{B}_2$ learns $r_i = \log_{g_i} c_i$ and $r = \log_h d$. Let $K = \mathsf{D}_{sk}(C)$. Our claim is that

$$\log K = \sum \omega_i r_i (x_i t + y_i) + \omega(z t + z') \; , \tag{14}$$

with $t = \mathsf{TCR}(C) \neq t^*$, is a uniform and independent element in $\mathbb{Z}_p$. Consider the set of linear equations over the hidden values $x_1, \ldots, x_n, y_1, \ldots, y_n, z, z'$ defined by Equations (12), (13), and (14), defined by the matrix $M \in \mathbb{Z}_p^{n+2 \times n+2}$,

$$M = \begin{array}{c} \begin{array}{cccccccc} x_1 & \ldots & x_n & y_1 & \ldots & y_n & z & z' \end{array} \\ \left(\begin{array}{cccccccc} \omega_1 & & & & & & \omega & \\ & \ddots & & & 0 & & \vdots & 0 \\ & & \omega_n & & & & \omega & \\ & & & \omega_1 & & & & \omega \\ & 0 & & & \ddots & & 0 & \vdots \\ & & & & & \omega_n & & \omega \\ \omega_1 r_1^* t^* & \cdots & \omega_n r_n^* t^* & \omega_1 r_1^* & \ldots & \omega_n r_n^* & \omega t^* r^* & \omega r^* \\ \omega_1 r_1 t & \cdots & \omega_n r_n t & \omega_1 r_1 & \ldots & \omega_n r_n & \omega t r & \omega r \end{array} \right) \end{array}$$

Since $\det(M) = \omega^2 \prod \omega_i (t - t^*)(\sum_{i=1}^n r_i - r)(\sum_{i=1}^n r_i^* - r^*) \neq 0$, Equation (14) is linearly independent of (12) and (13). ▌

6.5 A computational HPS based on Paillier

For a reader familiar with this concept we briefly sketch a computational hash-proof system based on Paillier's Decision Composite Residuosity (DCR) assumption [13]. For more details we refer the reader to [27, 13]. Let p_1, q_1, p_2, q_2 be primes where $p_1 = 2p_2 + 1$ and $q_1 = 2q_2 + 1$. Define $N_1 = p_1 q_1$ and $N_2 = p_2 q_2$. Consider

$$\mathbb{Z}^*_{N_1^2} = \mathbb{G}_{N_1} \times \mathbb{G}_{N_2} \times \mathbb{G}_2 \times T \; .$$

The subgroup $\mathbb{G} \subseteq \mathbb{Z}^*_{N_1^2}$ given by $\mathbb{G} = \mathbb{G}_{N_1} \times \mathbb{G}_{N_2}$ is cyclic of order $N_1 N_2$. Let g be a generator of $\mathbb{G}$. Then $g_1 = g^{N_2}$ is a generator of $\mathbb{G}_{N_1}$ and $g_2 = g^{N_1}$ is a generator of $\mathbb{G}_{N_2}$. Each element $h \in \mathbb{Z}_{N_1^2}$ can be uniquely written as $h = h_1 N_1 + h_2$, where $0 \leq h_1, h_2 < N_1$. We define $[h]_2 = h_2 = h \bmod N_1$.

Let N_1, g_2 be public parameters (that implicitly define the secret N_2). Define $\mathcal{C} = \mathbb{G}$ and $\mathcal{V} = \mathbb{G}_{N_2}$. Let $\mathsf{TCR} : \mathcal{C} \to \mathbb{Z}_{\lfloor N_1^2/2 \rfloor}$ be a target collision resistant hash function. For a $C = g_2^r \in \mathcal{V} \subset \mathcal{C}$, the element $r \in W' = \{0, \ldots, N_2 - 1\}$ is a witness (for $C \in \mathcal{V}$). Since N_2 is unknown, public sampling from the set $\mathcal{V}$ is done by picking random $r \in W = \{0, \ldots, \lfloor N_1/4 \rfloor\} \approx W'$ and computing $C = g_2^r$. Let $K = \{0, \ldots, \lfloor N_1^2/2 \rfloor\} \approx \{0, \ldots, N_1 N_2 - 1\}$. The set $\mathcal{S}$ is defined as $\mathcal{S} = \{(x, y) : x, y \in K\}$ and the projection $\mu : \mathcal{S} \to \mathcal{P}$ as $\mu(sk) = (u = g_2^x, v = g_2^y) \in \mathcal{P} = \mathbb{G}_{N_2}^2$. The hash function $\mathsf{D}_{sk} : \mathcal{C} \to \mathcal{K} = \mathbb{Z}_{N_1}$ is defined as $\mathsf{D}_{sk}(C) = [C^{xt+y} \bmod N_1^2]_2$, where $t = \mathsf{TCR}(C)$. Given witness $r \in W$ such that $C = g_2^r \in \mathcal{V}$ and $(u, v) = \mu(sk)$, public evaluation $\mathsf{HPS.pub}$ computes $K = \mathsf{D}_{sk}(C)$ as $K = [(u^t v)^r \bmod N_1^2]_2$. For correctness we refer to [13]. This completes the description of the hash-proof system $\mathscr{HPS}$.

As shown in [13], the subset membership problem is hard if the DCR assumption holds. Using the techniques from [13] it is now easy to show that if TCR is target collision resistant, then $\mathscr{HPS}$ is a computationally universal$_2$.

Acknowledgements

We thank Ronald Cramer, Jorge Villar, Chen Yuan, and Moti Yung for their comments. We thank Serge Vaudenay for pointing out the alternative definition of CCCA security.

References

[1] Masayuki ABE, Rosario Gennaro, and Kaoru Kurosawa. Tag-KEM/DEM: A new framework for hybrid encryption. Cryptology ePrint Archive, Report 2005/027, 2005. `http://eprint.iacr.org/`. (Cited on page 3.)

[2] Masayuki Abe, Rosario Gennaro, Kaoru Kurosawa, and Victor Shoup. Tag-KEM/DEM: A new framework for hybrid encryption and a new analysis of Kurosawa-Desmedt KEM. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 128–146. Springer-Verlag, Berlin, Germany, May 2005. (Cited on page 2, 3, 16.)

[3] M. Bellare, A. Boldyreva, K. Kurosawa, and Jessica Staddon. Multi-recipient encryption schemes: How to save on bandwidth and computation without sacrificing security. *IEEE Transactions on Information Theory*, ???(???), 2007. (Cited on page 17, 30.)

[4] Mihir Bellare, Tadayoshi Kohno, and Victor Shoup. Stateful public-key cryptosystems: How to encrypt with one 160-bit exponentiation. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 06*, pages 380–389. ACM Press, October / November 2006. (Cited on page 17.)

[5] Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In Tatsuaki Okamoto, editor, *ASIACRYPT 2000*, volume 1976 of *LNCS*, pages 531–545. Springer-Verlag, Berlin, Germany, December 2000. (Cited on page 11, 32, 33.)

[6] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93*, pages 62–73. ACM Press, November 1993. (Cited on page 2.)

[7] D. J. Bernstein. Pippenger's exponentiation algorithm. Available from `http://cr.yp.to/papers.html#pippenger`, 2001. (Cited on page 12.)

[8] Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity based encryption without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223–238. Springer-Verlag, Berlin, Germany, May 2004. (Cited on page 13, 16.)

[9] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55. Springer-Verlag, Berlin, Germany, August 2004. (Cited on page 3, 4, 18.)

[10] Xavier Boyen, Qixiang Mei, and Brent Waters. Direct chosen ciphertext security from identity-based techniques. In Vijayalakshmi Atluri, Catherine Meadows, and Ari Juels, editors, *ACM CCS 05*, pages 320–329. ACM Press, November 2005. (Cited on page 2, 17, 18, 30, 31, 32.)

[11] Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 207–222. Springer-Verlag, Berlin, Germany, May 2004. (Cited on page 2, 17.)

[12] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Hugo Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 13–25. Springer-Verlag, Berlin, Germany, August 1998. (Cited on page 1, 2, 16.)

[13] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 45–64. Springer-Verlag, Berlin, Germany, April / May 2002. (Cited on page 1, 2, 3, 16, 20, 21, 25, 26.)

[14] 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. (Cited on page 1, 2, 4, 5, 6, 8, 9, 11, 16, 17, 18, 20, 21, 30, 32, 33.)

[15] Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000. (Cited on page 1, 4.)

[16] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakley and David Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 10–18. Springer-Verlag, Berlin, Germany, August 1985. (Cited on page 16.)

[17] Rosario Gennaro and Victor Shoup. A note on an encryption scheme of Kurosawa and Desmedt. Cryptology ePrint Archive, Report 2004/194, 2004. http://eprint.iacr.org/. (Cited on page 2.)

[18] D. Hofheinz, J. Herranz, and E. Kiltz. The Kurosawa-Desmedt key encapsulation is not chosen-ciphertext secure. Cryptology ePrint Archive, Report 2006/207, 2006. http://eprint.iacr.org/. (Cited on page 2.)

[19] Antoine Joux and Kim Nguyen. Separating decision Diffie-Hellman from computational Diffie-Hellman in cryptographic groups. *Journal of Cryptology*, 16(4):239–247, September 2003. (Cited on page 2, 3.)

[20] Jonathan Katz and Moti Yung. Unforgeable encryption and chosen ciphertext secure modes of operation. In Bruce Schneier, editor, *FSE 2000*, volume 1978 of *LNCS*, pages 284–299. Springer-Verlag, Berlin, Germany, April 2000. (Cited on page 33.)

[21] E. Kiltz. Chosen-ciphertext secure key-encapsulation based on Gap Hashed Diffie-Hellman. In *Proceedings of PKC 2007*, volume 4450 of *LNCS*, pages 282 – 297, 2007. `http://eprint.iacr.org/2007/036`. (Cited on page 3, 4, 17, 18.)

[22] Eike Kiltz. Chosen-ciphertext security from tag-based encryption. In Shai Halevi and Tal Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 581–600. Springer-Verlag, Berlin, Germany, March 2006. (Cited on page 2, 4, 17.)

[23] Eike Kiltz. On the limitations of the spread of an IBE-to-PKE transformation. In Moti Yung, Yevgeniy Dodis, Aggelos Kiayias, and Tal Malkin, editors, *PKC 2006*, volume 3958 of *LNCS*, pages 274–289. Springer-Verlag, Berlin, Germany, April 2006. (Cited on page 2, 17.)

[24] Kaoru Kurosawa and Yvo Desmedt. A new paradigm of hybrid encryption scheme. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 426–442. Springer-Verlag, Berlin, Germany, August 2004. (Cited on page 2, 3, 11, 16, 19, 20, 21, 24.)

[25] Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In *21st ACM STOC*, pages 33–43. ACM Press, May 1989. (Cited on page 11, 17.)

[26] Digital signature standard, fips publication 186-3. National Institute of Standards and Technology, NIST FIPS PUB 186-3, U.S. Department of Commerce, March 2006. (Cited on page 30.)

[27] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Jacques Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 223–238. Springer-Verlag, Berlin, Germany, May 1999. (Cited on page 25.)

[28] Duong Hieu Phan and David Pointcheval. About the security of ciphers (semantic security and pseudo-random permutations). In Helena Handschuh and Anwar Hasan, editors, *SAC 2004*, volume 3357 of *LNCS*, pages 182–197. Springer-Verlag, Berlin, Germany, August 2004. (Cited on page 33.)

[29] Le Trieu Phong and Wakaha Ogata. On a variation of Kurosawa-Desmedt encryption scheme. Cryptology ePrint Archive, Report 2006/031, 2006. `http://eprint.iacr.org/`. (Cited on page 3, 17.)

[30] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 433–444. Springer-Verlag, Berlin, Germany, August 1992. (Cited on page 1, 4.)

[31] Phillip Rogaway, Mihir Bellare, John Black, and Ted Krovetz. OCB: A block-cipher mode of operation for efficient authenticated encryption. In *ACM CCS 01*, pages 196–205. ACM Press, November 2001. (Cited on page 2, 11.)

[32] Phillip Rogaway and Thomas Shrimpton. A provable-security treatment of the key-wrap problem. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 373–390. Springer-Verlag, Berlin, Germany, May / June 2006. (Cited on page 5.)

[33] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *22nd ACM STOC*, pages 387–394. ACM Press, May 1990. (Cited on page 17.)

[34] Hovav Shacham. A cramer-shoup encryption scheme from the linear assumption and from progressively weaker linear variants. Cryptology ePrint Archive, Report 2007/074, 2007. `http://eprint.iacr.org/`. (Cited on page 4.)

[35] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266. Springer-Verlag, Berlin, Germany, May 1997. (Cited on page 3, 18.)

[36] Victor Shoup. Using hash functions as a hedge against chosen ciphertext attack. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 275–288. Springer-Verlag, Berlin, Germany, May 2000. (Cited on page 1.)

[37] G. S. Vernam. Cipher printing telegraph systems for secret wire and radio telegraphic communications. *Journal of the American Institute of Electrical Engineers*, 45:109–115, 1926. (Cited on page 33.)

[38] Xiaoyun Wang and Hongbo Yu. How to break MD5 and other hash functions. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 19–35. Springer-Verlag, Berlin, Germany, May 2005. (Cited on page 17.)

[39] Xiaoyun Wang, Hongbo Yu, and Yiqun Lisa Yin. Efficient collision search attacks on SHA-0. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 1–16. Springer-Verlag, Berlin, Germany, August 2005. (Cited on page 17.)

[40] Mark N. Wegman and Larry Carter. New hash functions and their use in authentication and set equality. *Journal of Computer and System Sciences*, 22:265–279, 1981. (Cited on page 32.)

A An implicit-rejection variant of the dual KD scheme

We sketch a variant of our dual KD scheme from Section 4, where decapsulation is modified such that inconsistent ciphertexts get only implicitly rejected. Both variants have the same security properties as the dual KD scheme. We remark that it is also possible to give the same variant for the schemes based on the class of n-Linear assumptions from Section 5.

Let $\mathcal{GS}$ be a group scheme where $\mathcal{GR}_k$ specifies $(\hat{\mathbb{G}}, \mathbb{G}, g, p)$. We furthermore assume that $\mathcal{GR}_k$ contains a second random generator $h \in \mathbb{G}$. Let $\mathsf{TCR} : \mathbb{G} \rightarrow \mathbb{Z}_p$ be a target collision resistant hash function (for simplicity we assume TCR to be non-keyed). We build a key encapsulation mechanism $\mathcal{KEM} = (\mathsf{KEM.kg}, \mathsf{KEM.Enc}, \mathsf{KEM.Dec})$ with $\mathcal{K} = \mathbb{G}$ as follows.

```
KEM.Kg($1^k$)
    $x_1, x_2, y_2 \xleftarrow{\$} \mathbb{Z}_p^*$
    $u \leftarrow g^{-x_1/y_2}$ ; $v \leftarrow g^{-y_1/y_2} h^{1/y_2}$
    $pk \leftarrow (u, v) \in \mathbb{G}^2$
    $sk \leftarrow (x_1, x_2, y_1) \in (\mathbb{Z}_p)^3$
    Return $(sk, pk)$
```

```
KEM.Enc($pk$)
    $r \xleftarrow{\$} \mathbb{Z}_p^*$ ; $c \leftarrow g^r$
    $t \leftarrow \mathsf{TCR}(c)$ ; $\pi \leftarrow (u^t v)^r$
    $C \leftarrow (c, \pi) \in \mathbb{G}^2$
    $K \leftarrow h^r \in \mathbb{G}$
    Return $(C, K)$
```

```
KEM.Dec($sk, C$)
    Parse $C$ as $(c, \pi) \in \hat{\mathbb{G}} \times \hat{\mathbb{G}}$
    if $c \notin \mathbb{G}$ or $\pi \notin \mathbb{G}$ return $\perp$
    $t \leftarrow \mathsf{TCR}(c)$
    Return $K \leftarrow c^{x_1 t + y_1} \cdot \pi^{y_2}$
```

Note that decryption has to ensure that both elements c, π are contained in $\mathbb{G}$, whereas the explicit rejction scheme only has to check if c is contained in $\mathbb{G}$. Further, the two generators g, h can be viewed as fixed over a multi-user PKI environment. That means that g, h can be put in the general system parameters and only u, v have to be distinct for each user.

Correctness can be verified as follows. Without loss of generality, assume $(u^t v) \neq 1$. Let $\omega = \log_g h$. For an arbitrary ciphertext (c, π) we have $c = g^{r_1}$ and $\pi = (u^t v)^{r_2} = (g^{(-x_1/y_2)t - y_1/y_2} h^{1/y_2})^{r_2} = g^{r_2 \cdot \frac{-(x_1 t + y_1) + \omega}{y_2}}$, where $r_1 = r_2$ iff the ciphertext was correctly generated. Decapsulation computes K as

$$K = c^{x_1 t + y_1} \cdot \pi^{y_2} = g^{r_1(x_1 t + y_1) + r_2(-(x_1 t + y_1) + \omega)} = h^{r_2} \cdot g^{(r_1 - r_2)(x_1 t + y_1)}.$$

In case the ciphertext is consistent this reconstructs the correct key. In case the ciphertext is not consistent, decapsulation yields one uniform key K. However, in the IND-CCCA game it will be hard for an adversary to provide sufficient information (in form of a predicate) about K. Hence, informally speaking, inconsistent decapsulation queries are not useful.

A.1 Multi-user setting

The specific structure of the implicit-rejection KEM makes it possible to share the public elements g and h among many parties since decryption does not depend on the knowledge of $\omega = \log_g(h)$ anymore. Hence, similar to the Cramer-Shoup scheme, this implicit-rejection scheme can be used in the setting of multi-recipient encryption [3], where one single message is being simultaneously sent to a set of n different recipients.

Here the global system parameters consist of the two group elements g, h and the individual public key of the ith recipient only consists of the two group elements (u_i, v_i). To encrypt a message to n different recipient, the encapsulated symmetric key $K = h^r$ and the first element of the ciphertext $c = g^r$ can be shared among all recipients. Hence the multi-receiver ciphertext consists of $c = g^r$, the symmetric part $\mathsf{DEM.Enc}_K(M)$ and the individual parts $\pi_i = (u_i^t v_i)^r$, for $i \in \{1, \dots, n\}$. Compared to the naive solution (applying n-times the hybrid encryption scheme) this saves $n - 1$ times the symmetric part (each contains $m + k$ bits) plus n group elements.

B A hash-free variant of the dual KD scheme

Following [14] we give a hash-free variant $\mathcal{KEM} = (\mathsf{KEM.Kg}, \mathsf{KEM.Enc}, \mathsf{KEM.Dec})$ which can be used in case the dual KD KEM has to be implemented in groups where there is no efficient bijection $\mathsf{TCR}^* : \mathbb{G} \to \mathbb{Z}_p$ known. The hash-free variant basically implements an injective encoding $\mathsf{CHOP} : \hat{\mathbb{G}} \to \mathbb{Z}_p^\ell$ for a sufficiently large $\ell \geq 1$. In principle, such encodings always exist, since we can, similar to [14, 10], always write down the p-adic representation of any encoding of element $c \in \hat{\mathbb{G}}$. E.g., in case $\hat{\mathbb{G}} = \mathbb{Z}_{q'}$ and $\mathbb{G} \subseteq \mathbb{Z}_{q'}$ has prime-order p, one would have $\ell = \lceil \log_p(q') \rceil = \lceil |q'|/|p| \rceil$. For all practical choices of p and q' we have $q' \leq p^2$ (for example, all NIST recommended elliptic curves [26]) and hence $\ell \leq 2$. Hence, CHOP can be implemented using "mod p" operations at negliglible cost (compared to one exponentiation).

```
$\mathsf{KEM.Kg}(1^k)$
    $\omega \xleftarrow{\$} \mathbb{Z}_p^*$ ; $h \leftarrow g^\omega$
    $\forall i \in \{1, \dots \ell\} : u_i \xleftarrow{\$} g^{x_i}$
    $pk \leftarrow (u_1, \dots, u_\ell, h) \in \mathbb{G}^{\ell+1}$
    $sk \leftarrow (x_1, \dots, x_\ell, \omega) \in (\mathbb{Z}_p)^{\ell+1}$
    Return $(sk, pk)$
```

```
$\mathsf{KEM.Enc}(pk)$
    $r \xleftarrow{\$} \mathbb{Z}_p^*$ ; $c \leftarrow g^r$
    $(t_1, \dots, t_\ell) \leftarrow \mathsf{CHOP}(c)$
    $\pi \leftarrow (\prod_{i=1}^{\ell} u_i^{t_i})^r$
    $C \leftarrow (c, \pi) \in \mathbb{G}^2$
    $K \leftarrow h^r \in \mathbb{G}$
    Return $(C, K)$
```

```
$\mathsf{KEM.Dec}(sk, C)$
    Parse $C$ as $(c, \pi) \in \hat{\mathbb{G}} \times \hat{\mathbb{G}}$
    if $c \notin \mathbb{G}$ return $\perp$
    $(t_1, \dots, t_\ell) \leftarrow \mathsf{CHOP}(c)$
    if $c^{\sum_{i=1}^{\ell} x_i t_i} \neq \pi$ return $\perp$
    Return $K \leftarrow c^\omega$
```

Encryption takes $\ell+3$ exponentiations, where the generation of π can again be viewed as a single multi-exponentiation (as long as ℓ is a small constant). Decryption takes two exponentiations which can be viewed as one sequential exponentiation. Public-key contains $\ell+2$ elements in $\mathbb{G}$, secret-key $\ell+2$ element in $\mathbb{Z}_p$. Again, for most practical group schemes, $\ell = 2$.

Comparison with Kurosawa-Desmedt. A corresponding hash-free variant of the Kurosawa-Desmedt scheme has key sizes $|pk| = \ell'+2$, $|sk| = 2\ell'$, where $\ell' = \lceil 2|q'|/|p| \rceil$ (due to the fact that CHOP has to map two group elements to $\mathbb{Z}_p^{\ell'}$). Encryption has $\ell'+3$ exponentiations, decryption three. Assuming $q' \leq p^2$ we get $\ell' \leq 4$.

C Target collision resistant hashing

In the description of the schemes, the target collision resistent hash function TCR maps elements from $\mathbb{G}$ to $\mathbb{Z}_p$. However, $\mathbb{G}$ is a subgroup of $\hat{\mathbb{G}}$ and hence elements from $\mathbb{G}$ are usually represented as elements from $\hat{\mathbb{G}}$. Hence, what we need is a TCR function $\mathsf{TCR} : \hat{\mathbb{G}} \to \mathbb{Z}_p$ that is target collision-resistant on $\mathbb{G} \subseteq \hat{\mathbb{G}}$.

In this section we show that for many interesting group schemes $\mathcal{GS}$ we can implement such hash functions very efficiently by exploiting the fact that the order of $\mathbb{G}$ equals p, i.e. by giving a function $\mathsf{TCR}^* : \hat{\mathbb{G}} \to \mathbb{Z}_p$ that is a bijection on $\mathbb{G}$.

As already pointed out in [10] we note that it is sufficient for our application that TCR^* is injective on an overwhelming fraction of $\mathbb{G}$. In case we can efficiently find out if a given element is "non-bijective" we define TCR^+ as follows.

$$\mathsf{TCR}^+(c_1) = \left\{ \begin{array}{ccl} \perp & : & \text{if there exists } c_1' \neq c_1 \text{ with } \mathsf{TCR}^*(c_1) = \mathsf{TCR}^*(c_1') \\ \mathsf{TCR}^*(c_1) & : & \text{otherwise} \end{array} \right.$$

Our PKE schemes have to be adapted to handle the case that $\mathsf{TCR}^+(c_1)$ outputs $\perp$. For encryption, if $\mathsf{TCR}^+(c_1) = \perp$, then the encryption algorithm starts from scratch with a fresh (random) value $c_1 = g_1^r$. For decryption, all ciphertexts containing an element c_1 with $\mathsf{TCR}^+(c_1) = \perp$ simply get rejected.

Elliptic curves. An elliptic curve is defined by an equation of the form

$$y^2 = x^3 + ax + b \, .$$

If the coordinates x and y are chosen from a large finite field $\mathbb{F}_\ell$, the solutions form a finite abelian group $\hat{\mathbb{G}} = E(\mathbb{F}_\ell)$ with $\mathcal{O}$, the distinguished point at infinity, playing the role of multiplicative identity. According to Hasse's theorem the number of points on a curve is close to the size of the underlying field; more precisely, $(\sqrt{\ell} - 1)^2 \leq |E(\mathbb{F}_\ell)| \leq (\sqrt{\ell} + 1)^2$. We consider the case where $E(\mathbb{F}_\ell)$ already has prime order and set $\mathbb{G} = \hat{\mathbb{G}} = (\mathbb{F}_\ell)$. Then the mapping

$$\mathsf{TCR}^+(x, y) = \left\{ \begin{array}{ccl} \perp & : & x \geq \ell \\ x & : & \text{otherwise} \end{array} \right.$$

is a bijection on an overwhelming fraction of $\mathbb{G}$ [10]. We also remark that since $\hat{\mathbb{G}} = \mathbb{G}$, the subgroup membership test becomes trivial.

Group of Quadratic Residues Modulo Safe Prime. Let $\hat{\mathbb{G}} = \mathbb{Z}_{q'}$ for a safe prime q' with $q' = 2p+1$, where p is a prime. Let $\mathbb{G}$ be a group of nonzero quadratic residues modulo q'. The order of $\mathbb{G}$ equals p. Consider the following function

$$\mathsf{TCR}^*(x) = \begin{cases} x & : \text{ if } x \le p \\ q' - x & : \text{ otherwise.} \end{cases}$$

It is shown in [14, Example 2] that function TCR^* is a bijection. We remark that subgroup membership tests can be efficiently implemented by evaluating the Jacobi symbol.

D Construction of authenticated encryption schemes

We recall the encrypt-then-mac approach [5, 14] for constructing authenticated symmetric encryption.

D.1 Building blocks

Key Derivation Functions. A key-derivation function $\mathcal{KDF}$ for group scheme $\mathcal{GS}$ is a family of functions $\mathsf{KDF}_k : \mathbb{G} \to \{0,1\}^{2k}$. We assume its output on a random input is computationally indistinguishable from a random $2k$-bit string (pseudorandomness), captured by defining the kdf-pr-advantage of an adversary $\mathcal{B}_{\text{kdf}}$ as

$$\mathbf{Adv}^{\text{kdf-pr}}_{\mathcal{KDF},\mathcal{B}_{\text{kdf}}}(k) = \frac{1}{2} |\Pr[\mathcal{B}_{\text{kdf}}(\mathsf{KDF}(K)) = 1] - \Pr[\mathcal{B}_{\text{kdf}}(X) = 1]|,$$

where $K \xleftarrow{\$} \mathbb{G}$ and $X \xleftarrow{\$} \{0,1\}^{2k}$.

Message Authentication Codes. A message authentication code $\mathcal{MAC} = (\mathsf{M.tag}, \mathsf{M.vfy})$ with keys $mk \in \{0,1\}^k$ consists of a tag algorithm $\mathsf{M.tag}(mk, M)$ and a verification algorithm $\mathsf{M.vfy}(mk, \tau)$. For consistency we require that for all messages M, we have $\Pr[\mathsf{M.vfy}_{mk}(M, \mathsf{M.tag}(M)) \neq \bot] = 1$, where the probability is taken over the choice of coins of all the algorithms in the expression above.

$\mathcal{MAC}$ needs to be *strongly unforgeable against one-time attacks* (SUF-OT) captured by defining the suf-ot-advantage of an adversary $\mathcal{B}_{mac}$ as

$$\mathbf{Adv}^{\text{suf-ot}}_{\mathcal{MAC},\mathcal{B}_{mac}}(k) = \Pr[\mathsf{M.vfy}(mk, M^*, \tau^*) \neq \bot \ : \ mk \xleftarrow{\$} \{0,1\}^k \ ; \ (M^*, \tau^*) \xleftarrow{\$} \mathcal{B}_{mac}^{\mathsf{M.tag}(mk,\cdot)}(1^k)] \ .$$

Above, oracle $\mathsf{M.tag}(sk, \cdot)$ returns $\tau \leftarrow \mathsf{M.tag}(mk, M)$ and $\mathcal{A}$ may only make one single query to oracle $\mathsf{M.tag}(mk, \cdot)$. The target pair (M^*, τ^*) must be different from the pair (M, τ) obtained from $\mathsf{M.tag}(mk, \cdot)$ (strong unforgeability).

We remark that efficient MACs satisfying the above definition can be constructed without any computational assumption (and secure against unbounded adversaries) using, e.g., almost strongly-universal hash families [40].

One-time secure symmetric encryption. Symmetric encryption $\mathcal{S} = (\mathsf{S.Enc}, \mathsf{S.Dec})$ with keyspace $\{0,1\}^k$ and message space $\{0,1\}^{m(k)}$ is specified by its deterministic encryption algorithm $\mathsf{S.Enc}$ and decryption algorithm $\mathsf{S.Dec}$. The scheme needs to be IND-OT captured by defining the ind-ot-advantage $\mathbf{Adv}^{\text{ind-ot}}_{\mathcal{S},\mathcal{B}_s}(k)$ of an adversary $\mathcal{B}_s$ as

$$\mathbf{Adv}^{\text{ind-ot}}_{\mathcal{S},\mathcal{B}_s}(k) = \left| \Pr[b' = b : \ dk \xleftarrow{\$} \{0,1\}^k \ ; \ b \xleftarrow{\$} \{0,1\} \ ; \ b' \xleftarrow{\$} \mathcal{B}_s^{\mathsf{LoR}_b(\cdot,\cdot)}(1^k)] - \frac{1}{2} \right|$$

Above, $\mathsf{LoR}_b(M_0, M_1)$ returns $\psi \leftarrow \mathsf{S.Enc}(dk, M_b)$. $\mathcal{B}_s$ is allowed only one query to this left-or-right encryption oracle, consisting of a pair of equal-length messages. One example of an IND-OT secure symmetric encryption scheme is the one-time pad [37].

D.2 Construction of authenticated encryption

Let $\mathcal{S} = (\mathsf{S.Enc}, \mathsf{S.Dec})$ be a symmetric encryption that inputs keys from $\{0,1\}^k$ (such as AES), let $\mathcal{KDF}$ a key-derivation function for group scheme $\mathcal{GS}$ that outputs bitstrings of length $2k$, and let $\mathcal{MAC}$ be a MAC scheme with keys $mk \in \{0,1\}^k$. Using the "Encrypt-then-MAC" paradigm we can construct an algebraic $\mathcal{AE} = (\mathsf{AE.Enc}, \mathsf{AE.Dec})$ that inputs keys $K \in \mathbb{G}$ as follows.

```
$\mathsf{AE.Enc}(K, M)$
    $(mk||dk) \leftarrow \mathsf{KDF}(K)$, where $mk, dk \in \{0,1\}^k$
    $\psi' \leftarrow \mathsf{S.Enc}(dk, M)$
    $\tau \leftarrow \mathsf{M.tag}(mk, \psi')$
    Return $\psi = (\psi', \tau)$

$\mathsf{AE.Dec}(K, \psi = (\psi', \tau))$
    $(mk||dk) \leftarrow \mathsf{KDF}(K)$
    If $\mathsf{M.vfy}(mk, \tau) = \perp$ return $\perp$
    $M \leftarrow \mathsf{S.Dec}(dk, \psi')$
    Return $M$
```

Typically, a MAC tag (from a computationally secure MAC) has k bits, so the above construction generates ciphertexts of size $d(k) = |M| + k$. The following lemma [14, 20, 5] guarantees the AE scheme is one-time secure.

Lemma D.1 Assume $\mathcal{S}$ is IND-OT, $\mathcal{KDF}$ is pseudorandom, and $\mathcal{MAC}$ is SUF-OT. Then $\mathcal{AE}$ is AE-OT. In particlar, we have

$$\mathbf{Adv}^{ae\text{-ot}}_{\mathcal{AE},t}(k) \leq \mathbf{Adv}^{\text{ind-ot}}_{\mathcal{S},t}(k) + \mathbf{Adv}^{\text{kdf-pr}}_{\mathcal{KDF},t}(k) + \mathbf{Adv}^{\text{suf-ot}}_{\mathcal{MAC},t}(k) .$$

Intuitively, the MAC ensures the ciphertext integrity and the encrypt-then-mac paradigm ensure that one-time security is preserved.

We remark that for authenticated encryption is a strictly stronger security notion than chosen-ciphertext security (using a separation example from [5]), whereas the latter is already sufficient for the KEM/DEM composition theorem [14] (i.e., a IND-CCA secure KEM plus chosen-ciphertext secure symmetric encryption implies IND-CCA secure PKE). On the other hand, there exists redundancy-free chosen-ciphertext secure symmetric encryption [28] (with $d(k) = |M|$) whereas redundancy-free authenticated encryption do not exist.