KDM Security in the Hybrid Framework

Gareth T. Davies[*] and Martijn Stam

Department of Computer Science, University of Bristol,
Merchant Venturers Building, Woodland Road,
Bristol, BS8 1UB, United Kingdom.

{gareth.davies, martijn.stam}@bristol.ac.uk

Abstract

We study the natural question of how well suited the hybrid encryption paradigm is in the context of key-dependent message (KDM) attacks. We prove that if a key derivation function (KDF) is used in between the public (KEM) and symmetric (DEM) part of the hybrid scheme and this KDF is modelled as a random oracle, then one-wayness of the KEM and indistinguishability of the DEM together suffice for KDM security of the resulting hybrid scheme. We consider the most general scenario, namely CCA attacks and KDM functions that can call the random oracle. Although the result itself is not entirely unsuspected—it does solve an open problem from Black, Rogaway, and Shrimpton (SAC 2002)—proving it is considerably less straightforward; we develop some proof techniques that might be applicable in a wider context.

Keywords: KDM Security, Hybrid Encryption, KEM/DEM, Public Key Encryption.

[*] This author was partially supported by an EPSRC DTA award.

Contents

1 Introduction

When performing public key encryption (PKE) for large messages, it is often desirable to separate the encryption into two parts: public key techniques to encrypt a one-time symmetric key, and symmetric key techniques to encrypt the message. This type of encryption is commonly referred to as hybrid encryption. Hybrid encryption can be found in abundance in practice as it combines the benefits of flexible key management possible in the public key setting with the efficiency of symmetric encryption. Cramer and Shoup [27] were the first to capture hybrid encryption in a formal framework and their terminology is now commonly accepted: the public part of the algorithm is known as the key encapsulation mechanism (KEM), while the symmetric part, where the message is actually encrypted, is known as the data encapsulation mechanism (DEM). The main theorem of Cramer and Shoup (regarding hybrid encryption) is that the security properties of the KEM and the DEM can be regarded independently of each other. Loosely speaking, if any secure KEM is combined with any secure DEM, then the resulting public key encryption scheme is automatically secure. This theorem comes in two flavours, one for IND-CPA and one for IND-CCA2 security. Later works [39, 36, 2] looked at rebalancing the security properties of the KEM and the DEM such that the composition still guarantees IND-CCA2 security.

While, for good reasons, IND-CCA2 security has become the *de facto* security notion for public key encryption (so much so that we do not feel the need to elaborate on its definition here), there are situations where even this strongest of notions [14] is too weak. Particularly problematic is a situation where private keys themselves end up being encrypted. This is far from an academic question and the problem of secure encryption in the presence of key-dependent messages is becoming increasingly relevant. Applications of KDM security arise in disk encryption systems such as BitLocker [19], axiomatic security proofs [1, 4] and anonymous credential systems [24].

Encryption that is secure against key-dependent inputs was first studied by Abadi and Rogaway [1], who studied security proofs for protocols and showed how security is given by a reduction if no key-cycles exist in the protocol. Circular security was defined by Camenisch and Lysyanskaya [24] in the context of anonymous credentials. A more thorough treatment of key-dependent message (KDM) security was given by Black, Rogaway, and Shrimpton [18], who provided definitions of KDM security for both the public key and symmetric key setting. They proved that in the symmetric setting it is easy to achieve IND-KDM-CPA security in the random oracle model. For the public key setting they recall a simple scheme in the random oracle model [16], which they conjecture to be KDM secure but to the best of our knowledge, no proof of this has appeared in the literature. The scheme neatly fits the hybrid encryption framework, where the key encapsulation takes a random protokey r and applies a trapdoor one-way permutation to encapsulate it. At the same time, r is hashed (giving an ephemeral key) and XORed with the message (so the DEM is a one-time pad).[1] Thus a natural question arises: how well suited is the hybrid encryption paradigm in the context of key-dependent message attacks?

1.1 Our contribution.

Although many efforts have been made to study the security of PKE schemes under key-dependent messages, we provide the first treatment of hybrid encryption in this context. Before describing our result in detail, let us take a step back and look at the problem of KDM-secure hybrid encryption from a general perspective. For the standard (IND-CPA and IND-CCA) definitions it was possible to separate the security concerns of the KEM and the DEM. In a KDM setting it will be (a function of) the private key of the KEM that is encapsulated by the DEM using an ephemeral symmetric key. Intuitively, this can mean two things: either the appearance of the KEM's private key as input to the DEM ruins the normal separation of concerns, or the use of a freshly generated ephemeral key magically nullifies the key dependency. While it has been shown that KDM security does not follow from standard security definitions [3, 26], no explicit hybrid counterexamples have been given so far.

[1] Note that Black et al. assume that the random oracle returns as many bits as the message is long, which would require a slight modification to the original KEM-DEM framework.

It turns out constructing these counterexamples is relatively easy, implying that also in the hybrid setting dedicated constructions and proof methods are necessary to meet the strong KDM definition. Acar et al. [3] and Cash et al. [26] consider n-circular security and present schemes that fit standard security definitions, yet fail catastrophically in the presence of 2-cycles. In Section 4 we show that these schemes can be cast in the hybrid framework (in a way similar to how ElGamal encryption can be regarded as a hybrid encryption scheme) resulting in secure KEMs and DEMs, yet their combination is insecure in the presence of 2-cycles (due to the earlier works). We also show that the scheme of Acar et al. is not even secure against 1-cycles, whereas we suspect that the Cash et al. is under a suitable squared-Diffie-Hellman assumption. We even conjecture that the Cash et al. scheme is fully single-key KDM-secure in the generic group model (augmented with a pairing), which can be used to provide a black box separation between single query and multi query IND-KDM security notions (although breakdown of the standard hybrid is known, we are not aware of a prior explicit separation).

Our main result is presented in Section 3, where we consider the security of hybrid encryption schemes against key-dependent message attacks in the random oracle model. We show that if the key derivation function KDF is modelled as a random oracle, then a one-way secure (μOW-CCA) KEM and an indistinguishable (IND-CCA) DEM combine to form a PKE scheme that is (IND-KDM-CCA) secure against key-dependent message attacks, provided that the key-dependency does not involve the message length (this is a standard assumption). For our result, we make a distinction between KEMs for which there exists an efficient key-encapsulation–encapsulated-key checking oracle or not. Somewhat surprisingly, the former category gives a significantly tighter reduction, even though it means the KEM is weaker in some sense (in particular, it cannot be IND-CPA secure).

Although intuitively the random oracle would serve as a formidable barrier between the KEM and the DEM, removing any correlation, the proof turns out more involved than one might at first expect. To give a taster of the challenges, when reducing to a KEM security property, the simulation of valid DEM ciphertexts can be problematic without knowing the underlying message. If the DEM ciphertexts are uniformly distributed (over the randomness of the key) regardless of the message, simulation is easy (in particular, this observation suffices to prove the Black et al. scheme passively KDM secure). However, for arbitrary DEMs such simulation is not guaranteed and requires another game hop (to where a fixed message is encrypted). Luckily this does not lead to a circularity (where a DEM hop requires a KEM hop to set up, which itself requires a DEM hop to go through etc.).

Indeed, in our proof, we use the well-known identical-until-bad technique in a way similar to Dent's analysis of IND-CCA secure KEMs in the random oracle model [28]. However, a crucial innovation in our proof is a novel use of the "deferred analysis" technique of [31] to analyse the bad event: it turns out that we need to bound the events in two different games, but in one of these, the key-dependency hinders the usual approaches. Our solution is to move the analysis of the bad event to the other game, where the analysis is considerably easier. In contrast with the original deferred analysis, the probability of the events changes, so we need to account for this game-hop separately.

Technically even more challenging is the reduction to the DEM's indistinguishability, since the key dependency function needs to be mapped to a message for the DEM. A complication arises in that an adversary making multiple challenge queries could let the message to-be-encrypted by the DEM depend on prior, ephemeral DEM keys. We introduce a new security notion for the DEM that captures this type of key dependency (namely on past keys only) and show that is equivalent to standard IND-CCA. Moreover, we show how to map key dependency functions in the PKE world to this restricted set of key dependency functions in the DEM world by modelling the random oracle as a pseudorandom function. Thus, rather bizarrely, our security bound for KDM security of hybrid encryption includes a PRF term, despite there not being a PRF in the construction itself.

1.2 Related Work.

Efforts in the area of KDM security have focused on either positive results giving circular secure schemes, or negative impossibility results. On the positive side, Boneh et al. [19] gave KDM-CPA scheme secure under the Decisional Diffie-Hellman assumption, a result strengthened by Camenisch et al. [23] to a KDM-CCA scheme. Many other positive results have also been presented [40, 4, 35, 9, 10, 37, 41]. Many public key schemes have been proposed of a number theoretical nature, where the class of functions for which KDM security can be proven is related, typically in some algebraic sense, to the scheme itself (see [12, 20, 21, 6] and the references contained therein).

There have also been negative results, and in particular Haitner and Holenstein [34] showed the impossibility of obtaining KDM security based on standard assumptions and using standard techniques, and also separation results of Acar et al. [3] and Cash et al. [26].

In [7] a comparatively efficient scheme is given based on the LWE/LPN problems. Recent work has also looked at circular security in the context of point obfuscation [25], identity-based encryption [5] and bit-encryption [44]. The development of fully homomorphic encryption by Gentry [32] utilises encryption of the secret key under the corresponding public key, and recent work of Brakerski and Vaikuntanathan has looked at KDM security with FHE in more detail [22]. In the symmetric setting, recent work has focused on authenticated encryption [15], and Bellare et al. [13] describe ciphers that can securely encipher their own keys. In [47] Unruh presents a new definition, PROG-KDM security, which combines KDM security and key corruptions in the same definition.

2 Preliminaries

2.1 Notation.

If x is a string then $|x|$ denotes the length of x, and $x||y$ denotes the concatenation of strings x and y. If S is a finite set then $|S|$ is its cardinality and $s \overset{\$}{\leftarrow} S$ denotes picking s uniformly at random from S. A property of a boolean variable, which we will call a *flag*, is that once `true` it stays `true`. Boolean flags are assumed initialized to `false`. The adversary, which we regard as code of a program, makes calls to the oracles, taking as input values from some finite domain associated to each oracle.

In our proofs we will make extensive use of the game-hopping technique. To do this we use the notation of Bellare and Rogaway [17] and the ideas of Shoup [46]. The adversary and game outputs can be regarded as random variables. We write $\mathbf{Pr}\left[\mathsf{G}^{\mathcal{A}} = 1\right]$ for the probability that the game output is 1 when we run game G with adversary $\mathcal{A}$. Games G_i and G_j are *identical until* bad if their code differs only in statements that follow the setting of Boolean flag bad to `true`. We will use the following, fundamental lemma of game playing [17]:

Lemma 2.1. Let G_i, G_j be *identical until* bad games, and let $\mathcal{A}$ be an adversary. Then

$$\left|\mathbf{Pr}\left[\mathsf{G}_i^{\mathcal{A}} = 1\right] - \mathbf{Pr}\left[\mathsf{G}_j^{\mathcal{A}} = 1\right]\right| \leq \mathbf{Pr}\left[\mathsf{G}_j^{\mathcal{A}} \text{ sets } \mathsf{bad}\right] .$$

Definition 2.2. [Pseudorandom functions] Let $F : \mathcal{I} \times \mathcal{D} \to \mathcal{R}$ be a family of functions from domain $\mathcal{D}$ to range $\mathcal{R}$ indexed by seeds $\mathcal{I}$.

For $x \in \mathcal{I}$ we let $F_x(y) : \mathcal{D} \to \mathcal{R}$ be defined by $F_x(y) = F(x, y)\ \forall y \in \mathcal{D}$.

Let $Fun[\mathcal{D}, \mathcal{R}]$ be the set of all functions from $\mathcal{D}$ to $\mathcal{R}$. Set $\mathcal{D} = \{0,1\}^\lambda$ for some security parameter λ.

Then the PRF advantage of an adversary $\mathcal{A}$ attacking F is given by

$$\mathbf{Adv}_{F,\mathcal{A}}^{\mathsf{PRF}}(\lambda) \overset{def}{=} \mathbf{Pr}\left[x \overset{\$}{\leftarrow} \mathcal{I} : \mathcal{A}^{F_x(\cdot)} = 1\right] - \mathbf{Pr}\left[g \overset{\$}{\leftarrow} Fun[\{0,1\}^\lambda, \mathcal{R}] : \mathcal{A}^{g(\cdot)} = 1\right] . \quad (1)$$

2.2 Public Key and Hybrid Encryption

We briefly recall the syntax of a public key encryption scheme PKE, consisting of four algorithms Pg, Kg, Enc, and Dec. Parameter generation Pg takes as input a security parameter λ and outputs a set of parameters common among multiple keypairs (e.g. the description of an elliptic curve); key generation Kg takes the parameters and outputs a public–private key pair $(\mathrm{pk}, \mathrm{sk})$; encryption Enc takes as input the public key and a message from $\{0,1\}^*$ (or some other message space with a well-defined

```
$\underline{\mathsf{Hyb.Pg}(1^\lambda)}$
$\mathrm{pars}_{\mathsf{KEM}} \leftarrow \mathsf{KEM.Pg}(1^\lambda)$
$\mathrm{pars}_{\mathsf{DEM}} \leftarrow \mathsf{DEM.Pg}(1^\lambda)$
**return** $(\mathrm{pars}_{\mathsf{KEM}}, \mathrm{pars}_{\mathsf{DEM}})$

$\underline{\mathsf{Hyb.Kg}(\mathrm{pars})}$
$(\mathrm{pk}, \mathrm{sk}) \leftarrow \mathsf{KEM.Kg}(\mathrm{pars}_{\mathsf{KEM}})$
**return** $(\mathrm{pk}, \mathrm{sk})$
```

```
$\underline{\mathsf{Hyb.Enc}(\mathrm{pars}, \mathrm{pk}, m)}$
$(K, C) \leftarrow \mathsf{KEM.encap}_{\mathrm{pk}}()$
$h_K \leftarrow \mathsf{KDF}_{\mathrm{pars},\mathrm{pk}}(K)$
$\psi \leftarrow \mathsf{DEM.Enc}_{h_K}(m)$
**return** $(C, \psi)$
```

```
$\underline{\mathsf{Hyb.Dec}(\mathrm{pars}, \mathrm{sk}, C, \psi)}$
$K \leftarrow \mathsf{KEM.decap}_{sk}(C)$
$h_K \leftarrow \mathsf{KDF}_{\mathrm{pars},\mathrm{pk}}(K)$
$m \leftarrow \mathsf{DEM.Dec}_{h_K}(\psi)$
**return** $m$
```

Figure 1: *Construction of a hybrid cryptosystem* Hyb.

length measure) and outputs a ciphertext; decryption Dec takes as input the private key and a purported ciphertext and returns a message in $\{0,1\}^*$ or some designated error symbol $\perp$. The standard security notions for public key encryption are indistinguishability (IND) under chosen plaintext attacks (CPA), respectively chosen ciphertext attacks (CCA). We refer to e.g. Bellare et al.[14] for formal definitions.

A popular way of constructing public key schemes is through the use of hybrid encryption, consisting of a key encapsulation mechanism $\mathsf{KEM} = (\mathsf{KEM.Pg}, \mathsf{KEM.Kg}, \mathsf{KEM.encap}, \mathsf{KEM.decap})$, a data encapsulation mechanism (DEM) $\mathsf{DEM} = (\mathsf{DEM.Pg}, \mathsf{DEM.Enc}, \mathsf{DEM.Dec})$, and often a key derivation function KDF as compatibility layer in between, as depicted in Fig. 1. We use the term *protokey* to describe the input to the KDF (above denoted K). The individual components have the following properties.

- The KEM's parameter and key generation work as for a public key encryption scheme. Key encapsulation $\mathsf{KEM.encap}$ takes a public key and returns both a key $K \in \mathcal{K}_{\mathsf{KEM}}$ and an encapsulation C thereof. Key decapsulation $\mathsf{KEM.decap}$ takes as input a private key and a purported key encapsulation and returns a key in $\mathcal{K}_{\mathsf{KEM}}$ or some designated error symbol $\perp$.

- Data encapsulation $\mathsf{DEM.Enc}$ takes a message $m \in \{0,1\}^*$ and a symmetric key in $\mathcal{K}_{\mathsf{DEM}}$ and outputs an encryption ψ. A data decapsulation $\mathsf{DEM.Dec}$ takes a message encapsulation ψ and a symmetric key in $\mathcal{K}_{\mathsf{DEM}}$ and outputs the message m or error symbol $\perp$.

- A key derivation function KDF is a deterministic algorithm implementing a mapping from $\mathcal{K}_{\mathsf{KEM}}$ to $\mathcal{K}_{\mathsf{DEM}}$. Note that in addition to some key K the algorithm takes as input $\mathsf{KEM}.\mathrm{pk}$ and $\mathsf{DEM}.\mathrm{pars}$ (in order to determine $\mathcal{K}_{\mathsf{KEM}}$ and $\mathcal{K}_{\mathsf{DEM}}$).

As a notational convention, we omit parameters and implicitly assume that they are fed to every algorithm, and we write key inputs as subscripts except in cases where the operation really is on the key. In the single user setting, we often write $(\mathrm{pk}, \mathrm{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$ as shorthand for running parameter and key generation in one go.

While hybrid encryption has been in widespread use ever since the advent of public key cryptosystems, the first formalisation of the paradigm was given by Cramer and Shoup [27]. They gave security definitions of IND-CPA and IND-CCA security for both the KEM and the DEM part and proved that in the standard model, where the key derivation function only needs to be (close to) balanced, the public key cryptosystem inherits security from its constituent parts, e.g. IND-CCA security for both the KEM and the DEM part is a sufficient condition to obtain an IND-CCA secure hybrid PKE scheme. Since then efforts have been made [39, 36, 2] to investigate how weakening the individual security notions impacts on the security of the PKE scheme. We refer to the above-mentioned articles for general security notions for KEMs and DEMs; in Appendix A we list the two notions, μOW-CCA (where the μ indicates there can be multiple key pairs in the game) and IND-CCA, that are relevant for this paper.

The event $\mathsf{Coll}_{\mathsf{KEM}}(q_{\mathsf{LR}}, \lambda)$, paramaterised by the number of oracle queries the adversary makes and the security parameter, implies a collision in the ephemeral key output by the KEM, which is extremely unlikely to occur (if it were, this would also adversely affect the KEM's one-wayness).

Dent [28] looked at various constructions of KEMs from one-way secure public key cryptosystems (operating on a restricted message space). He modelled the key derivation function as a random oracle and considered it as part of the KEM. A typical example of such a construction is the use of a trapdoor function to encapsulate a protokey r that is subsequently hashed to arive at derived key $\mathsf{H}(r)$ (this is the same KEM as mentioned in the introduction). He shows several elegant, generic KEM constructions that are IND-CCA secure based on fairly minimal assumptions on the encryption scheme used to encrypt the protokey. For instance, in the example above security is attained if the trapdoor function is one-way secure even in the presence of an oracle that checks whether a ciphertext is a valid ciphertext or not (i.e., the actual range of the trapdoor function is easily recognizable by the adversary), which Dent calls OW-CPA+ security. If the KEM is constructed from a *randomized* public key cryptosystem, security based on one-wayness is proven, provided that there is an efficient plaintext-ciphertext checking oracle, that, when given a message and ciphertext pair, correctly determines whether the ciphertext is an encryption of the message or not.

Our results are reminiscent of that of Dent, however whereas he exploits Cramer–Shoup's composition theorem and only explicitly considered the construction of secure KEMs that incorporate the KDF, we (necessarily) look at hybrid encryption as a whole and our emphasis will be on constructing secure hybrid cryptosystems from a KEM treating the KDF separately. Thus where Dent used a public key encryption scheme to arrive at a protokey, we use a proper key encapsulation mechanism. As a consequence, our framework is on the one hand more general then Dent's (e.g. we can deal with Diffie–Hellman type KEMs more easily), yet on the other we are likely to run into similar technicalities. In particular, we see a dichotomy in the KEMs depending on the availability of a key-encapsulation–encapsulated-key checking oracle $\mathsf{KEM.Check}_{\mathrm{pk}}(C, K)$ that, on input a key encapsulation C and purported encapsulated (proto) key K decides whether $\mathsf{KEM.decap}_{\mathrm{sk}}(C) = K$ or not. This leads to the following two types of KEMs, and each type will give a different reduction in the security analysis further on:

- In TYPE-1 KEMs there is an efficient checking oracle $\mathsf{KEM.Check}_{\mathrm{pk}}(C, K)$. This class encompasses all schemes that determine the encapsulation C deterministically based on the key K, including the usual schemes based on trapdoor permutations/functions. Diffie–Hellman type KEMs in a pairing-based setting (where DDH is easy) can also be part of this class. (Looking ahead, in the security proof when an adversary makes a query $\mathsf{H}(K)$ to its random oracle, the checking oracle allows the reduction to determine whether this K corresponds to some challenge encapsulation C.)

- In TYPE-2 KEMs there is no efficient checking oracle. This class contains all IND-CPA secure KEMs. (Again looking ahead, the lack of a checking oracle means that the reduction will need to guess whether a query $\mathsf{H}(K)$ corresponds to a challenge ciphertext or not, leading to a less tight reduction.)

2.3 Key Dependent Message (KDM) Security

The first formal definition of KDM security was given by Black et al. [18]. They define a KDM analogue IND-KDM-CPA of the established IND-CPA security notion. Simply put, an adversary submits as challenge a function φ and receives either an encryption of $\varphi(\mathrm{sk})$ or of a dummy message $0^{|\varphi(sk)|}$. Camenisch et al. [23] introduced the "active" version IND-KDM-CCA security as a natural blend between IND-CCA and IND-KDM-CPA, and this is the version we will focus on. There can be multiple keys in the system and, contrary to standard IND-CPA security, for the IND-KDM security notions it is not possible to reduce (e.g. by hybrid argument) to a single key or single query.

The IND-KDM notions are relative to a function class Φ, which stipulates that the adversary is bound to asking only queries $\varphi \in \Phi$. For instance, if Φ corresponds to the set of all constant functions, notions equivalent to IND-CPA and IND-CCA emerge. The challenge is to devise schemes that can be proven secure for an as large as possible class Φ. Black et al. formally regarded φ modelled as an algorithm in some fixed RAM model; furthermore they imposed length-regularity of φ in the sense that $|\varphi(\mathrm{sk})|$ does not depend on the value sk. Following e.g. Unruh [47], we will instead regard φ as an arithmetic or Boolean circuit, which will imply that the output length of φ is fixed (and automatically independent of its input).

```
$\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA}[\Phi]\text{-}b}_{\mathsf{PKE},\,\mathcal{A}}(\lambda)$:
    $\mathrm{pars} \leftarrow \mathsf{Pg}(1^\lambda)$
    $t \leftarrow 0$
    $\mathsf{FL} \leftarrow \emptyset$
    $\mathbf{sk} \leftarrow ()$
    $b' \leftarrow \mathcal{A}^{\mathsf{New},\mathsf{LR}_b,\mathsf{Dec}}(\mathrm{pars})$
    return $b'$

$\mathsf{New}()$:
    $t \leftarrow t+1$
    $(\mathrm{pk}_t, \mathrm{sk}_t) \leftarrow \mathsf{Kg}(\mathrm{pars})$
    Append $\mathrm{sk}_t$ to $\mathbf{sk}$
    return $\mathrm{pk}_t$

$\mathsf{LR}_b(\varphi, i)$:
    if $\varphi \notin \Phi(\mathrm{pars}, \mathbf{pk}, i)$ then
        return $\lightning$
    $m_1 \leftarrow \varphi(\mathbf{sk})$
    $m_0 \leftarrow 0^{|m_1|}$
    $\tilde{C}_b \leftarrow \mathsf{Enc}_{\mathrm{pk}_i}(m_b)$
    $\mathsf{FL} \leftarrow \mathsf{FL} \cup \{(\tilde{C}_b, i)\}$
    return $\tilde{C}_b$

$\mathsf{Dec}(C, i)$:
    if $(C, i) \in \mathsf{FL}$ then
        return $\lightning$
    $m \leftarrow \mathsf{Dec}_{\mathrm{sk}_i}(C)$
    return $m$
```

Figure 2: *The general* IND-KDM-CCA *experiment for public key encryption. The bit b is hard-wired into the Left-or-Right oracle* LR_b *and determines whether a key-dependent message or a dummy is encrypted and returned to $\mathcal{A}$. Removing oracle* Dec *yields the* IND-KDM-CPA *experiment.*

Our syntax also differs from that of Black et al. as we make a distinction between parameter and key generation, which is not uncommon in multi-user settings. Since φ implements a function from a Cartesian product of secret key spaces to the message space and these spaces can depend on the parameter generation (e.g. which cyclic group is used for DLP based systems), the security experiment incorporates a check that φ is syntactically valid (however, we will henceforth drop explicit mention of it).

Definition 2.3. Let PKE $=$ (Gen, Enc, Dec) be a public key encryption scheme (with security parameter λ). Let Φ be a collection of circuits that map a (number of) secret key(s) to an element in the message space. Then the IND-KDM-atk$[\Phi]$ advantage of an adversary $\mathcal{A}$ against PKE relative to key-dependent message attacks for circuit class Φ and atk $\in$ {CPA, CCA} is defined by

$$\mathbf{Adv}^{\mathsf{IND\text{-}KDM\text{-}atk}[\Phi]}_{\mathsf{PKE},\,\mathcal{A}}(\lambda) \overset{def}{=} \left| \mathbf{Pr}\left[\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}atk}[\Phi]\text{-}1}_{\mathsf{PKE},\,\mathcal{A}}(\lambda) = 1\right] - \mathbf{Pr}\left[\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}atk}[\Phi]\text{-}0}_{\mathsf{PKE},\,\mathcal{A}}(\lambda) = 1\right]\right|$$

where the experiment $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA}[\Phi]\text{-}b}_{\mathsf{PKE},\,\mathcal{A}}(\lambda)$ is given in Fig. 2, and removing the decryption oracle yields experiment $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CPA}[\Phi]\text{-}b}_{\mathsf{PKE},\,\mathcal{A}}(\lambda)$.

2.3.1 Alternative Definitions and Special Cases.

We will use variants of this definition in this paper but will retain this format for consistency. A special case of KDM security is the 1-cycle case of the wider field of circular security, a framework in which a scheme remains secure under a so-called 'key-cycle', where we have a cycle of public/secret key-pairs $(\mathrm{pk}_i, \mathrm{sk}_i)$ for $i = 1, ..., t$, and each sk_i is encrypted under $\mathrm{pk}_{(i \bmod t)+1}$.

Backes et al. [10, 9] used a different framework where the adversary does not have direct access to the results of encryptions but instead can instruct the system to create keys, perform encryptions and other operations etc. with the subsequent capacity to learn part of the system's state. This is a potentially stronger framework (see also Dent [29]) reminiscent of work on cryptographic APIs (e.g. [38]). The latest such definition (the PROG-KDM security definition provided by Unruh [47]) also allows for corruptions, but it is not easy to see how it can be satisfied in a non-programmable random oracle setting (let alone the standard model). In this work we will contend ourselves with the easier (and weaker) notion based on the original work by Black et al.

3 IND-KDM-CCA Security of Hybrid Encryption

3.1 Restricted KDM Security of the DEM

We introduce a security notion for DEMs called IND-PKDM-CCA ('Prior-KDM'), where an adversary's KDM capability is restricted to (encryptions of) functions of all 'past' DEM keys in the system. The formal security game for IND-PKDM-CCA is depicted in Fig. 3. Our reductions for KDM security of hybrid encryption will use this IND-PKDM-CCA security notion for the DEM. However, by a hybrid argument one can show that this restricted form of KDM attacks is not all that useful to an attacker—the notion is in fact equivalent to IND-CCA security (see Appendix B).

```
$\mathbf{Exp}^{\mathsf{IND\text{-}PKDM\text{-}CCA\text{-}}b}_{\mathsf{DEM},\ \mathcal{A}}(\lambda)$:
  $\mathrm{pars} \leftarrow \mathsf{DEM.Pg}(1^\lambda)$
  $i \leftarrow 0$
  $\mathsf{FL} \leftarrow \emptyset$
  $b' \leftarrow \mathcal{A}^{\mathsf{New},\mathsf{LR}_b,\mathsf{Dec}}(\mathrm{pars})$
  return $b'$
```

```
$\mathsf{New}()$:
  $i \leftarrow i + 1$
  $K_i \leftarrow \mathsf{DEM.Kg}(\mathrm{pars})$
  return $i$
```

```
$\mathsf{LR}_b(j, \vartheta)$:
  if $j \notin [i]$ then
    return $\lightning$
  $m_1 \leftarrow \vartheta(K^{j-1})$
  $m_0 \leftarrow 0^{|m_1|}$
  $\psi \leftarrow \mathsf{DEM.Enc}_{K_j}(m_b)$
  $\mathsf{FL} \leftarrow \mathsf{FL} \cup \{(j, \psi)\}$
  return $\psi$
```

```
$\mathsf{Dec}(j, \psi)$:
  if $(j, \psi) \in \mathsf{FL}$ then
    return $\lightning$
  $m \leftarrow \mathsf{DEM.Dec}_{K_j}(\psi)$
  return $m$
```

Figure 3: *The* IND-PKDM-CCA *security experiment for data encapsulation mechanism* DEM. *Here* $\vartheta(K^{i-1})$ *indicates the function* ϑ *can depend on all keys in range* $\{K_0, ..., K_{i-1}\}$.

3.2 Hybrid Encryption is IND-KDM-CCA Secure (in the ROM)

Let Hyb = (Hyb.Gen, Hyb.Enc, Hyb.Enc) be a hybrid encryption scheme and let $\mathcal{A}$ be an adversary. In the hybrid setting there are two types of keys present: the private key of the KEM and the ephemeral key for the DEM, where knowledge of the private KEM key leads to immediate recovery of the ephemeral key. When we regard Hyb as a public key encryption scheme in the context of key-dependent messages, it follows from Fig. 1 that it is on the private key of the *KEM* that key-dependent messages (that are input to the *DEM*) will depend. For concreteness, in Fig. 4 we have expanded Fig. 2 in the context of hybrid encryption where the key derivation function is modelled as a random oracle. The forbidden list FL ensures that the adversary cannot trivially win.

We show that any KEM/DEM system that has a TYPE-1 μOW-CCA KEM and an IND-CCA DEM gives a IND-KDM-CCA[Φ] secure hybrid encryption scheme provided that the key derivation function KDF is modelled as a random oracle, and the functions in Φ can call the random oracle. By this we mean that when modelled as circuits, $\varphi \in \Phi$ can have gates that explicitly call the random oracle. Here, the μ indicates that there is a choice of multiple targets to invert. Recall that our modelling of functions in $\varphi \in \Phi$ as circuits implicitly implies that φ is length-regular, meaning that given pk and φ, one can uniquely determine the length of $\varphi(\mathrm{sk})$ (this is the same restriction as made by Black et al. [18] and Backes et al. [9]). This result is formalized in Theorem 3.1. In Theorem C.1 we provide an analogous, but significantly less tight result for TYPE-2 KEMs.

Proof intuition. In our proof we make use of the game-playing technique [46, 17] and introduce a sequence of games, as described in Fig. 5, and the games themselves are specified in Fig. 6. Apart from the simple, syntactical transitions (2) and (3), there are five game-hops to bound $\mathcal{A}$'s advantage distinguishing $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA\text{-}1}}_{\mathsf{Hyb},\ \mathcal{A}}(\lambda)$ and $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA\text{-}0}}_{\mathsf{Hyb},\ \mathcal{A}}(\lambda)$. These are denoted with solid lines. Here (4) and (5) are identical-until-bad hops. The 'Forbidden List' FL ensures that the adversary cannot win trivially. We define bad to be the event that the adversary queries the random oracle on a protokey K previously used by the left-or-right oracle.

So far, this is all standard fair: use the security of the KEM to decouple the key encapsulated by the KEM and the one used by the DEM (where Dent [28] used the same bad event in his analysis of IND-CCA secure KEMs), followed by a straightforward indistinguishability hop to the DEM. Unfortunately, with the introduction of key-dependent messages the latter hop has become quite a bit more burdensome; moreover bounding the bad event in the presence of key-dependent messages is somewhat troublesome. To overcome these challenges, our proof uses a number of techniques. To invoke the DEM's indistinguishability, the standard reduction would pick all the KEM keypairs and use these to simulate the KEM part of the hybrid encryption scheme (to run the adversary against the entire PKE). Since the reduction itself is playing the DEM indistinguishability game, it can use its DEM oracles to complete the DEM part (as the protokey encapsulated by the KEM and the ephemeral key used by the DEM are decoupled at this point). However if an adversary (against the PKE) may make queries with

```
$\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA}\text{-}b}_{\mathsf{Hyb},\,\mathcal{A}}(\lambda)$:
    $\text{pars} \leftarrow \mathsf{Pg}(1^\lambda)$
    $t \leftarrow 0$
    $\mathbf{sk} \leftarrow ()$
    $\mathsf{H}_{\mathsf{LIST}} \leftarrow \emptyset$
    $\mathsf{FL} \leftarrow \emptyset$
    $b' \leftarrow \mathcal{A}^{\mathsf{New},\mathsf{H},\mathsf{LR}^{\mathsf{H}}_b,\mathsf{Dec}^{\mathsf{H}}}(\text{pars})$
    return $b'$

$\mathsf{New}()$:
    $t \leftarrow t+1$
    $(\text{pk}_t, \text{sk}_t) \leftarrow \mathsf{Kg}(\text{pars})$
    Append $\text{sk}_t$ to $\mathbf{sk}$
    return $\text{pk}_t$

$\mathsf{H}$:
    On query $K$:
    if $(K, h_K) \in \mathsf{H}_{\mathsf{LIST}}$
        return $h_K$
    else
        $h_K \xleftarrow{\$} \{0,1\}^\lambda$
        $\mathsf{H}_{\mathsf{LIST}} \leftarrow \mathsf{H}_{\mathsf{LIST}} \cup \{(K, h_K)\}$
        return $h_K$

$\mathsf{LR}^{\mathsf{H}}_b(\varphi, i)$:
    if $\varphi \notin \Phi(\text{pars}, \mathbf{pk}, i)$ then
        return $\lightning$
    $m_1 \leftarrow \varphi^{\mathsf{H}}(\mathbf{sk})$
    $m_0 \leftarrow 0^{|m_1|}$
    $(C, K) \leftarrow \mathsf{KEM.encap}_{\text{pk}_i}()$
    $h_K \leftarrow \mathsf{H}(K)$
    $\psi_b \leftarrow \mathsf{DEM.Enc}_{h_K}(m_b)$
    $\mathsf{FL} \leftarrow \mathsf{FL} \cup \{(C, \psi_b, i)\}$
    return $(C, \psi_b)$

$\mathsf{Dec}^{\mathsf{H}}(C, \psi, i)$:
    if $(C, \psi, i) \in \mathsf{FL}$ then
        return $\lightning$
    $K \leftarrow \mathsf{KEM.decap}_{\text{sk}_i}(C)$
    if $K = \perp$ then
        return $\perp_{\mathsf{KEM}}$
    $h_K \leftarrow \mathsf{H}(K)$
    $m \leftarrow \mathsf{DEM.Dec}_{h_K}(\psi)$
    if $m = \perp$ then
        return $\perp_{\mathsf{DEM}}$
    return $m$
```

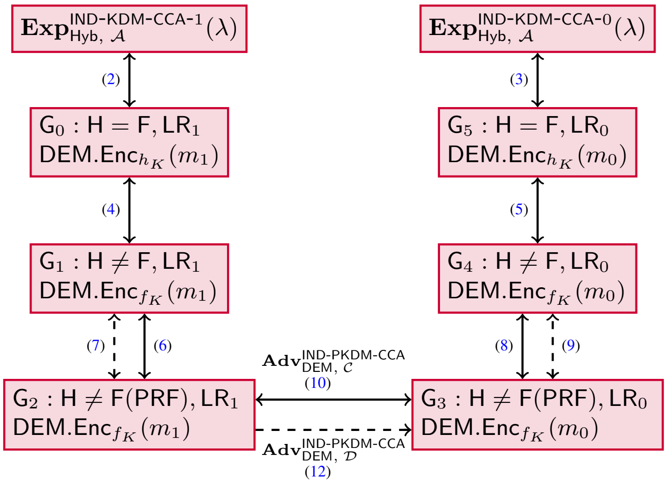
Figure 4: *The* IND-KDM-CCA *indistinguishability experiment made explicit for multi-key hybrid encryption in the random oracle model. The adversary is allowed to query decryptions of the challenge ciphertexts under different public keys than the ones generated by* LR_b*, and this restriction is dealt with by the list* FL.

KDM functions that call the random oracle, it could in principle submit functions that decrypt past key encapsulations and, with the help of the random oracle, turn them in past DEM keys (effectively, the KDM function can cause the event that would normally have triggered bad). Since the reduction does not know the actual DEM keys being used, it suddenly finds itself in a tight spot and a direct hybrid argument (to get rid of past DEM keys) does not seem to work.

Our solution is to leverage the newly introduced IND-PKDM-CCA notion. Since we model the KDM functions as circuits, it turns out to be possible to describe a compiler that turns a KDM function against the PKE into one against the DEM. There is however one further complication. For the public key scheme, we model the hash function as a random oracle and *the KDM function has access to the random oracle*. Yet, for the DEM scheme there is no random oracle present, which would suggest that the KDM function in the DEM world should not depend on one either. Moreover, it is not possible to predict on which values the KDM function would call the random oracle. Thus, when the random oracle is implemented by the reduction using lazy sampling, though it could hard-code the hash list so far into the circuit, the simulation might fail once fresh values are requested. To handle this, we (partly) model the random oracle as a pseudorandom function (rather than using lazy sampling). This provides the reduction a succinct description of the *entire* random oracle and it can safely embed the key to the pseudorandom function in the circuit used in the IND-PKDM-CCA game. The introduction of a PRF requires two additional hops (6) and (8).

The bounding of event bad is relatively easy on the m_0-side of the diagram, as one does not need to know the KEM's private key sk in order to simulate the data encapsulations: bad is bounded in G_3 by $\mathbf{Adv}^{\mu\mathsf{OW\text{-}CCA}}_{\mathsf{KEM},\,\mathcal{B}}$. However, on the m_1-side of the diagram it is less obvious how to bound the bad event, since it is not possible to simulate the key-dependent values. The solution is to move the bad event from the m_1-side to the m_0-side using the separate hop (12), which bounds the difference between $\mathbf{Pr}[\text{bad}]$ in games G_2 and G_3. This incurs a second $\mathbf{Adv}^{\mathsf{IND\text{-}PKDM\text{-}CCA}}_{\mathsf{DEM},\,\mathcal{C}}$ term to the bound.

Bounding of the bad event breaks down if distinct queries to the LR oracle made identical KDF queries. We bound this event by the separate quantity $\mathsf{Coll}_{\mathsf{KEM}}(q_{\mathsf{LR}}, \lambda)$. It might be possible to avoid this technicality by changing the scheme so it hashes $\mathsf{H}(C, K)$ instead of just $\mathsf{H}(K)$.



$\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA\text{-}1}}_{\mathsf{Hyb},\,\mathcal{A}}(\lambda)$
(2)
$\mathsf{G}_0 : \mathsf{H} = \mathsf{F}, \mathsf{LR}_1$
$\mathsf{DEM.Enc}_{h_K}(m_1)$
(4)
$\mathsf{G}_1 : \mathsf{H} \neq \mathsf{F}, \mathsf{LR}_1$
$\mathsf{DEM.Enc}_{f_K}(m_1)$
(7)
(6)
$\mathsf{G}_2 : \mathsf{H} \neq \mathsf{F}(\mathsf{PRF}), \mathsf{LR}_1$
$\mathsf{DEM.Enc}_{f_K}(m_1)$
$\mathbf{Adv}^{\mathsf{IND\text{-}PKDM\text{-}CCA}}_{\mathsf{DEM},\,\mathcal{C}}$
(10)
$\mathbf{Adv}^{\mathsf{IND\text{-}PKDM\text{-}CCA}}_{\mathsf{DEM},\,\mathcal{D}}$
(12)
$\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA\text{-}0}}_{\mathsf{Hyb},\,\mathcal{A}}(\lambda)$
(3)
$\mathsf{G}_5 : \mathsf{H} = \mathsf{F}, \mathsf{LR}_0$
$\mathsf{DEM.Enc}_{h_K}(m_0)$
(5)
$\mathsf{G}_4 : \mathsf{H} \neq \mathsf{F}, \mathsf{LR}_0$
$\mathsf{DEM.Enc}_{f_K}(m_0)$
(8)
(9)
$\mathsf{G}_3 : \mathsf{H} \neq \mathsf{F}(\mathsf{PRF}), \mathsf{LR}_0$
$\mathsf{DEM.Enc}_{f_K}(m_0)$

Game	Oracle	Oracle Model	Message
$\mathbf{Exp}^1$	H	LS	m_1
G_0	$\boxed{\mathsf{H} = \mathsf{F}}$	LS	m_1
G_1	$\boxed{\mathsf{H} \neq \mathsf{F}}$	LS	m_1
G_2	$\mathsf{H} \neq \mathsf{F}$	$\boxed{\text{PRF}}$	m_1
G_3	$\mathsf{H} \neq \mathsf{F}$	PRF	$\boxed{m_0}$
G_4	$\mathsf{H} \neq \mathsf{F}$	$\boxed{\text{LS}}$	m_0
G_5	$\boxed{\mathsf{H} = \mathsf{F}}$	LS	m_0
$\mathbf{Exp}^0$	$\boxed{\mathsf{H}}$	LS	m_0

Figure 5: *Diagrammatic overview of the game hopping structure of proof that an* μOW-CCA TYPE-1 *secure KEM and an* IND-CCA *secure DEM yield a* IND-KDM-CCA *secure hybrid scheme in the random oracle model. The games are defined formally in Fig. 6, here the boxes indicate the game* G_i *and which oracles comprise the game. The transitions are labelled by the equations in the proof. The table indicates which oracle is used to handle* $\mathcal{A}$*'s calls, how we model the random oracle (LS denotes lazy sampling) and which message is encrypted. The boxed items indicate where a change occurs in the hop from one game to another.*

Interpretation. When it comes to hybrid schemes, our result is very general. Indeed, it even generalizes the work by Dent [28] (restricted to IND-CPA-security) as we can deal with key encapsulation schemes where the protokey is derived from the randomness in a hard-to-invert fashion. For instance, if $\mathbb{G}_p$ is a cyclic group of order p with generator g, an obvious Diffie–Hellman inspired KEM would pick private key $x \in \mathbb{Z}_p^*$, set public key g^x and compute a key encapsulation by generating a random $r \in \mathbb{Z}_p^*$, releasing g^r as the encapsulation of $K = g^{rx}$. Our theorems can deal with this situation (where the KEM is TYPE-1 iff DDH is easy in $\mathbb{G}_p$), but it is not covered by the KEMs given by Dent.

Black et al. [18] suggest the use of a variant of TDP-KEM combined with a one-time pad as a KDM-secure public key scheme in the random oracle model. Here TDP-KEM is shorthand for trapdoor-permutation-KEM, where the public and private key of the KEM match that of the trapdoor permutation and key encapsulation takes a random K in the domain of the trapdoor permutation, applies the permutation to encapsulate and outputs $\mathsf{H}(K)$ as ephemeral key, or, in the hybrid model with explicit key derivation function (Fig. 4) the KEM would output K as ephemeral protokey.

As a result of our theorem, if we restrict this scheme to any fixed-size message length, security is guaranteed. Strictly speaking, for arbitrary length messages, we would need to allow signalling of (an upper bound on) the message length to the random oracle so it can output the required number of bits. This is primarily a syntactical issue that we did not feel sufficiently important to incorporate into our main framework. Since TDP-KEM has an obvious checking oracle, we regard our Theorem 3.1 settling the problem left open by Black et al.

Theorem 3.1. Let Hyb be a hybrid PKE scheme (Fig. 1) with a TYPE-1 KEM, with the key derivation function modelled by a random oracle. Let Φ be any set of functions, including those which have random oracle access. Let F be a family of pseudorandom functions. Then for any adversary $\mathcal{A}$ calling LR at most q_{LR} times, there exists algorithms $\mathcal{B}$ and $\mathcal{C}$ (of comparable computational complexity) such that

$$\mathbf{Adv}^{\mathsf{IND\text{-}KDM\text{-}CCA}[\Phi]}_{\mathsf{Hyb},\,\mathcal{A}}(\lambda) \leq 2\mathbf{Adv}^{\mu\mathsf{OW\text{-}CCA}}_{\mathsf{KEM},\,\mathcal{B}}(\lambda) + 2\mathbf{Adv}^{\mathsf{IND\text{-}PKDM\text{-}CCA}}_{\mathsf{DEM},\,\mathcal{C}}(\lambda) + 2\mathsf{Coll}_{\mathsf{KEM}}(q_{\mathsf{LR}}, \lambda) + 4\mathbf{Adv}^{\mathsf{PRF}}_{F,\,\mathcal{A}}(\lambda)\,.$$

This theorem, combined with Theorem B.1, yields the following corollary relating to standard definitions.

Corollary 3.2. As above, and let n be the number of DEM keys in the system, then:

$$\mathbf{Adv}^{\mathsf{IND\text{-}KDM\text{-}CCA}[\Phi]}_{\mathsf{Hyb},\,\mathcal{A}}(\lambda) \leq 2\mathbf{Adv}^{\mu\mathsf{OW\text{-}CCA}}_{\mathsf{KEM},\,\mathcal{B}}(\lambda) + 2n\cdot\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{DEM},\,\mathcal{C}}(\lambda) + 2\mathsf{Coll}_{\mathsf{KEM}}(q_{\mathsf{LR}}, \lambda) + 4\mathbf{Adv}^{\mathsf{PRF}}_{F,\,\mathcal{A}}(\lambda)\,.$$

Proof. [of Theorem 3.1]

We recall that Fig. 4 contains a description of the security games $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA}\text{-}b}_{\mathsf{Hyb},\ \mathcal{A}}(\lambda)$ that are obtained by specifying the general PKE IND-KDM-CCA games for hybrid encryption where the key derivation function is modelled by a random oracle H. (For simplicity, we omit explicit mention of the class Φ in the description of the security experiments.) As is customary, we use lazy sampling to define H's behaviour, maintaining a list $\mathsf{H}_{\mathsf{LIST}}$ of query pairs (K, h_K) produced by H so far.

```
$\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA}[\Phi]\text{-}b}_{\mathsf{PKE},\ \mathcal{A}}(\lambda)$:
  $\mathrm{pars} \leftarrow \mathsf{Pg}(1^\lambda)$
  $t \leftarrow 0$
  $\mathbf{sk} \leftarrow ()$
  $\mathsf{H}_{\mathsf{LIST}}, \mathsf{F}_{\mathsf{LIST}}, \mathsf{FL} \leftarrow \emptyset$
  $b' \leftarrow \mathcal{A}^{\mathsf{New},\mathsf{H},\mathsf{LR}_b,\mathsf{Dec}}(\mathrm{pars})$
  return $b'$

$\mathsf{LR}_b(\varphi^{\mathsf{H}}, i)$:
  if $\varphi^{\mathsf{H}} \notin \Phi(\mathrm{pars}, \mathbf{pk}, i)$ then
      return $\lightning$
  $m_1 \leftarrow \varphi(\mathbf{sk})$
  $m_0 \leftarrow 0^{|m_1|}$
  $(C, K) \leftarrow \mathsf{KEM.encap}_{\mathrm{pk}_i}()$
  $h_K \leftarrow \mathsf{F}(K)$
  $\psi_b \leftarrow \mathsf{DEM.Enc}_{h_K}(m_b)$
  $\mathsf{FL} \leftarrow \mathsf{FL} \cup \{(C, \psi_b, i)\}$
  return $(C, \psi_b)$

$\mathsf{New}()$:
  $t \leftarrow t + 1$
  $(\mathrm{pk}_t, \mathrm{sk}_t) \leftarrow \mathsf{Kg}(\mathrm{pars})$
  Append $\mathrm{sk}_t$ to $\mathbf{sk}$
  return $\mathrm{pk}_t$
```

```
$\mathsf{H}(K)$:
  if $(K, h_K) \in \mathsf{H}_{\mathsf{LIST}}$
      return $h_K$
  if $(K, h_K) \in \mathsf{F}_{\mathsf{LIST}}$
      set bad $\leftarrow$ true
      return $h_K$
  $h_K \xleftarrow{\$} \{0,1\}^\lambda$
  $\mathsf{H}_{\mathsf{LIST}} \leftarrow \mathsf{H}_{\mathsf{LIST}} \cup \{(K, h_K)\}$
  return $h_K$

$\mathsf{F}(K)$:
  if $(K, h_K) \in \mathsf{F}_{\mathsf{LIST}}$
      return $h_K$
  if $(K, h_K) \in \mathsf{H}_{\mathsf{LIST}}$
      set bad $\leftarrow$ true
      return $h_K$
  $h_K \xleftarrow{\$} \{0,1\}^\lambda$
  $\mathsf{F}_{\mathsf{LIST}} \leftarrow \mathsf{F}_{\mathsf{LIST}} \cup \{(K, h_K)\}$
  return $h_K$
```

```
$\mathsf{HF}(K)$:
  if $(K, h_K) \in \mathsf{F}_{\mathsf{LIST}}$
      return $h_K$
  if $(K, h_K) \in \mathsf{H}_{\mathsf{LIST}}$
      return $h_K$
  $h_K \xleftarrow{\$} \{0,1\}^\lambda$
  $\mathsf{H}_{\mathsf{LIST}} \leftarrow \mathsf{H}_{\mathsf{LIST}} \cup \{(K, h_K)\}$
  return $h_K$

$\mathsf{Dec}(C, \psi, i)$:
  if $(C, \psi, i) \in \mathsf{FL}$ then
      return $\lightning$
  call $K \leftarrow \mathsf{Decap}(C, i)$
  if $K = \perp$ then
      return $\perp_{\mathsf{KEM}}$
  $h_K \leftarrow \mathsf{HF}(K)$
  $m \leftarrow \mathsf{DEM.Dec}_{h_K}(\psi)$
  if $m = \perp$ then
      return $\perp_{\mathsf{DEM}}$
  return $m$
```

```
$\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA}[\Phi]\text{-}b}_{\mathsf{PKE},\ \mathcal{A}}(\lambda)$:
  $\mathrm{pars} \leftarrow \mathsf{Pg}(1^\lambda)$
  $t \leftarrow 0$
  $x \xleftarrow{\$} \{0,1\}^\lambda$
  $\mathbf{sk} \leftarrow ()$
  $\mathsf{H}_{\mathsf{LIST}}, \mathsf{F}_{\mathsf{LIST}}, \mathsf{FL} \leftarrow \emptyset$
  $b' \leftarrow \mathcal{A}^{\mathsf{New},\mathsf{H},\mathsf{LR}_b,\mathsf{Dec}}(\mathrm{pars})$
  return $b'$
```

```
$\mathsf{H}(K)$:
  if $(K, h_K) \in \mathsf{H}_{\mathsf{LIST}}$
      return $h_K$
  if $(K, h_K) \in \mathsf{F}_{\mathsf{LIST}}$
      set bad $\leftarrow$ true
  $h_K \leftarrow \mathsf{PRF}_x(K)$
  $\mathsf{F}_{\mathsf{LIST}} \leftarrow \mathsf{F}_{\mathsf{LIST}} \cup \{(K, h_K)\}$
  return $h_K$
```

```
$\mathsf{HF}(K)$:
  if $(K, h_K) \in \mathsf{F}_{\mathsf{LIST}}$
      return $h_K$
  if $(K, h_K) \in \mathsf{H}_{\mathsf{LIST}}$
      return $h_K$
  $h_K \leftarrow \mathsf{PRF}_x(K)$
  $\mathsf{H}_{\mathsf{LIST}} \leftarrow \mathsf{H}_{\mathsf{LIST}} \cup \{(K, h_K)\}$
  return $h_K$
```

Figure 6: *Security games used for proof Theorem 3.1. Games* G_2 *and* G_3 *are described below the line, with all items the same as above apart from these changes. Games* G_0 *and* G_5 *correspond to the code including the highlighting, implying that* $\mathsf{H} = \mathsf{F}$ *(as far as I/O behaviour is concerned). Games* $\mathsf{G}_1 - \mathsf{G}_4$ *have* $\mathsf{H} \neq \mathsf{F}$ *as two independently sampled random oracles. Games* G_2 *and* G_3 *model the random oracle as a PRF, rather than using lazy sampling. Games* G_0*,* G_1 *and* G_2 *correspond to* $b = 1$*, whereas Games* G_3*,* G_4 *and* G_5 *correspond to* $b = 0$*.*

In the game there are four distinct places where queries to H could be made. Firstly, the adversary $\mathcal{A}$ can make direct H queries; any query to the oracle LR_b will require one 'direct' call to H for the key derivation and may include a number of indirect calls as part of the specified function φ; and finally as a decryption query for key derivation. For the purpose of our game-hopping approach, we need to be able to make a clear distinction between these cases. To this end, we introduce two additional oracles: F and HF. We make a syntactical change so that LR_b always uses F for its key derivation, and Dec always uses

HF. Oracle HF synchronises with items that are added to lists for both H and F. By ensuring that F, H and HF implement the same random oracle (i.e. are functionally equivalent, exhibiting exactly the same input/output behaviour), the changed games are equivalent to the original security experiments.

In Fig. 6, G_0 corresponds to such a modified, yet equivalent game, in this case for $b = 1$. The $b = 0$ sibling game is called G_5. In both these games the oracles H and F each maintain their own list, $\mathsf{H}_{\mathrm{LIST}}$, respectively $\mathsf{F}_{\mathrm{LIST}}$, yet control code ensures *(a)* that these two lists can not contain K overlap in the sense that no triple (K, h_K, h'_K) can exist for which both $(K, h_K) \in \mathsf{H}_{\mathrm{LIST}}$ and $(K, h'_K) \in \mathsf{F}_{\mathrm{LIST}}$ and *(b)* that the oracles H and F will look up elements from the other oracle's list, thus ensuring synchronisation. As a result of this design, F and H are functionally equivalent to each other in the games G_0 and G_5, implying that from an adversary's point of view G_0 is equivalent to $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA\text{-}1}}_{\mathsf{Hyb},\,\mathcal{A}}(\lambda)$, or

$$\mathbf{Pr}\left[\mathsf{G}_0{}^{\mathcal{A}} = 1\right] = \mathbf{Pr}\left[\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA\text{-}1}}_{\mathsf{Hyb},\,\mathcal{A}}(\lambda) = 1\right] . \tag{2}$$

Similarly we claim that G_5 is equivalent to $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA\text{-}0}}_{\mathsf{Hyb},\,\mathcal{A}}(\lambda)$, so

$$\mathbf{Pr}\left[\mathsf{G}_5{}^{\mathcal{A}} = 1\right] = \mathbf{Pr}\left[\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA\text{-}0}}_{\mathsf{Hyb},\,\mathcal{A}}(\lambda) = 1\right] . \tag{3}$$

We proceed by a more interesting hop, where we make F and H independent. The oracles F and H are modified such that when a query is made to one oracle (say H) that has previously been queried to the other (F) then a fresh value is still created (and added to $\mathsf{H}_{\mathrm{LIST}}$). Moreover, in this case the flag bad is set to `true` first. This is described in Fig. 6, where the new G_1 corresponds to the $b = 1$ case and G_4 to the $b = 0$ case. By syntactical inspection, G_0 and G_1 are identical up to the point at which the flag is set, enabling application of the fundamental lemma of game-hopping:

$$\left|\mathbf{Pr}\left[\mathsf{G}_0{}^{\mathcal{A}} = 1\right] \;-\; \mathbf{Pr}\left[\mathsf{G}_1{}^{\mathcal{A}} = 1\right]\right| \leq \mathbf{Pr}\left[\mathcal{A}\text{ sets bad in }\mathsf{G}_1\right] \tag{4}$$

and in a similar vein G_4 and G_5 are identical until bad, so

$$\left|\mathbf{Pr}\left[\mathsf{G}_4{}^{\mathcal{A}} = 1\right] \;-\; \mathbf{Pr}\left[\mathsf{G}_5{}^{\mathcal{A}} = 1\right]\right| \leq \mathbf{Pr}\left[\mathcal{A}\text{ sets bad in }\mathsf{G}_4\right] . \tag{5}$$

(To bound the difference between games G_4 and G_5 a standard hop involving the KEM's IND-CCA advantage is an alternative.)

The hop between the key-dependent scenario and the non-key-dependent world will be problematic later on due to the fact that if φ calls the random oracle, the simulation cannot correctly answer these queries since it does not know the values of the DEM keys in the system, only their indices. To counter this we add two additional hops in which we use a PRF rather than lazy sampling to model our random oracle. We regard the φ that acts on sk (of the KEM) as a circuit, with some gates that call the RO. Thus there is a (one-to-one) mapping from φ circuits (which act on sk) to ϑ circuits (that act on the DEM keys). We assume that there is some kind of 'safe storage' of all DEM keys. In this manner it is possible to track the past RO queries that are made by these ϑ functions.

These H gates will have some inputs, and will check if the input string corresponds to some $\mathsf{H}_{\mathrm{LIST}}$ entry, or an $\mathsf{F}_{\mathrm{LIST}}$ entry. If it is an F query, then assign a K_i to some of the output wires (since the game does not know the K_i but it can use them). The issue, however, is that if $\mathcal{A}$ gives a circuit φ that makes an H query in a gate, and subsequently makes another H query then the $\mathsf{H}_{\mathrm{LIST}}$ lists will not be synchronised.

To counter this, consider H as a pseudorandom function $\mathsf{PRF} : \{0,1\}^\lambda \times \mathcal{K}_{\mathsf{DEM}} \to \{0,1\}^\lambda$ chosen from some PRF-secure function family F, parameterised by some seed $x \in \{0,1\}^\lambda$, rather than using lazy sampling. Denote $\mathsf{PRF}_x(K)$ as being the PRF applied to input K and seed x. The gates for H now store the $\mathsf{F}_{\mathrm{LIST}}$, and when calls to F are made we can wire up the corresponding K_i values. When the function makes H calls, we simply implement the PRF on the given input. To make this subtle change, we need to implement another two (symmetrical) game hops in which we change the way we model the random oracle from lazy sampling (LS) to using a PRF. The difference between $\mathcal{A}$'s advantage against G_1 and its advantage against G_2 is bounded by $\mathcal{A}$'s advantage in breaking the PRF:[2]

$$\begin{aligned}\mathbf{Pr}\left[\mathsf{G}_1{}^{\mathcal{A}} = 1\right] - \mathbf{Pr}\left[\mathsf{G}_2{}^{\mathcal{A}} = 1\right] &\leq \mathbf{Adv}^{\mathsf{PRF}}_{F,\,\mathcal{A}}(\lambda) &(6)\\ \mathbf{Pr}\left[\mathcal{A} \text{ sets bad in } \mathsf{G}_1\right] - \mathbf{Pr}\left[\mathcal{A} \text{ sets bad in } \mathsf{G}_2\right] &\leq \mathbf{Adv}^{\mathsf{PRF}}_{F,\,\mathcal{A}}(\lambda) &(7)\end{aligned}$$

and likewise the difference between $\mathcal{A}$'s advantage against G_3 and its advantage against G_4 is bounded by the PRF advantage:

$$\begin{aligned}\mathbf{Pr}\left[\mathsf{G}_3{}^{\mathcal{A}} = 1\right] - \mathbf{Pr}\left[\mathsf{G}_4{}^{\mathcal{A}} = 1\right] &\leq \mathbf{Adv}^{\mathsf{PRF}}_{F,\,\mathcal{A}}(\lambda) &(8)\\ \mathbf{Pr}\left[\mathcal{A} \text{ sets bad in } \mathsf{G}_4\right] - \mathbf{Pr}\left[\mathcal{A} \text{ sets bad in } \mathsf{G}_3\right] &\leq \mathbf{Adv}^{\mathsf{PRF}}_{F,\,\mathcal{A}}(\lambda) &(9)\end{aligned}$$

Now we are in a position to consider the hop between games G_2 and G_3. In game G_2 the response from the Left-or-Right oracle is given to the adversary by LR_1, resulting to an encryption of $m_1 = \varphi(\mathbf{sk})$ in $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA\text{-}1}}_{\mathsf{Hyb},\,\mathcal{A}}(\lambda)$, whereas in game G_3, the Left-or-Right oracle is implemented by LR_0, leading to an encryption of $m_0 = 0^{|\varphi(\mathbf{sk})|}$ (as in $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA\text{-}0}}_{\mathsf{Hyb},\,\mathcal{A}}(\lambda)$). To show that games G_2 and G_3 are distinguishable only with small probability we introduce an adversary $\mathcal{C}$ that attacks the IND-PKDM-CCA property of the DEM, and show that as long as the DEM is secure in this respect, then the output of the games is indistinguishable. More precisely,

$$\mathbf{Pr}\left[\mathsf{G}_2{}^{\mathcal{A}} = 1\right] - \mathbf{Pr}\left[\mathsf{G}_3{}^{\mathcal{A}} = 1\right] \leq \mathbf{Adv}^{\mathsf{IND\text{-}PKDM\text{-}CCA}}_{\mathsf{DEM},\,\mathcal{C}}(\lambda) \qquad (10)$$

The consequence of F and H being independently sampled oracles is that in games G_2 and G_3 the encapsulated key and the key used for the DEM are effectively decoupled (as the adversary has no direct access to F). This decoupling allows us to use a DEM hop to prove equation (10), as shown by the reduction in Fig. 7. In the game that $\mathcal{C}$ plays, it runs $\mathcal{A}$ as a black-box that returns a valid φ, then $\mathcal{C}$ creates messages m_0 and m_1 in the same way that the LR_b oracle does in the other games. However, where in the games G_2 and G_3 there was an explicit oracle F that provided linkage between a key K output by the KEM and its corresponding key h_K actually used by the DEM, in the simulation $\mathcal{C}$ uses its own oracles to create the keys h_K in the IND-PKDM-CCA experiment it itself is playing. To do this, we need to move the function φ that acts on the KEM secret keys to the function ϑ, that acts upon DEM keys. The set K_ϑ contains all the DEM keys that are currently in the system. To simulate the DEM hop we need to make sure that the ϑ circuit in the IND-PKDM-CCA game is consistent with the circuit that acts on all of the DEM keys in the system in the PKE game. Every time $\mathcal{A}$ makes an F query in its PKE game we need to add that key to the set of keys that ϑ can act upon.

In this decoupled scenario, reduction $\mathcal{C}$ generates the $(\mathrm{pk}, \mathrm{sk})$ pairs itself. The seed of the PRF is then 'hardwired' into the gates of ϑ so when $\mathcal{A}$'s KDM function makes a RO call, it is dealt with by this setup. This allows the simulation to go through without $\mathcal{C}$ actually knowing which values K_i are queried to the RO. The messages m_1 and m_0 are then 'created' just as they are in $\mathcal{A}$'s LR queries. Now $\mathcal{D}$ calls its own oracles LR, New and Dec (in the IND-PKDM-CCA game) and returns a pair (C, ψ_b) as $\mathcal{A}$ would have expected.

The LR oracle in the simulation translates the φ into a ϑ. If this function makes an oracle call K_i, the simulation checks $\mathsf{H}_{\mathsf{LIST}}$ for an entry containing K_i, and if present returns the corresponding h_K. If the value is on $\mathsf{F}_{\mathsf{LIST}}$ then the simulation will know the index of the key but not the value itself, and thus a PRF gate can be called to retrieve the corresponding h_K. If it is on neither list, simply initiate PRF on K_i.

Since the adversary $\mathcal{A}$ has no direct access to F this indirect simulation of F is perfect. As a result, if $\mathcal{C}$ is in $\mathbf{Exp}^{\mathsf{IND\text{-}PKDM\text{-}CCA\text{-}1}}_{\mathsf{DEM},\,\mathcal{C}}(\lambda)$ then $\mathcal{A}$ will behave towards $\mathcal{C}$ exactly as it would do in G_2, and similarly if $\mathcal{C}$ is in $\mathbf{Exp}^{\mathsf{IND\text{-}PKDM\text{-}CCA\text{-}0}}_{\mathsf{DEM},\,\mathcal{C}}(\lambda)$ then $\mathcal{A}$ will behave as in G_3, proving (10).

[2] The more usual hop in a proof would be to replace a pseudorandom function by a perfectly random function, whereas here the perfect object is substituted by a computational approximation—for bounding the difference between the two worlds the 'direction' is irrelevant.

```
$\mathcal{C}$ playing $\mathbf{Exp}^{\mathsf{IND\text{-}PKDM\text{-}CCA}\text{-}b}_{\mathsf{DEM},\,\mathcal{C}}(\lambda)$:
  $\mathrm{pars} \leftarrow \mathsf{KEM.Pg}(1^\lambda)$
  $\mathbf{sk} \leftarrow ()$
  $\mathsf{F}_{\mathrm{LIST}} \leftarrow \emptyset$
  $\mathsf{H}_{\mathrm{LIST}} \leftarrow \emptyset$
  $\mathsf{FL} \leftarrow \emptyset$
  $x \overset{\$}{\leftarrow} \{0,1\}^\lambda$
  $b' \leftarrow \mathcal{A}^{\mathsf{LR}_b,\mathsf{New},\mathsf{H},\mathsf{Dec}}(\mathrm{pk})$
  return $b'$

New():
  $t \leftarrow t+1$
  $(\mathrm{pk}_t, \mathrm{sk}_t) \leftarrow \mathsf{Kg}(\mathrm{pars})$
  Append $\mathrm{sk}_t$ to $\mathbf{sk}$
  return $\mathrm{pk}_t$
```

```
$\mathsf{LR}_b(\varphi^{\mathsf{H}}, n)$:
  $\varphi^{\mathsf{H}} \rightarrow \vartheta$
  if $\varphi^{\mathsf{H}}$ makes RO call $K_i$ then
      $h_K \leftarrow \mathsf{PRF}_x(K)$
      $K_\vartheta \leftarrow K_\vartheta \cup K_i$
  $m_1 \leftarrow \vartheta(K_\vartheta)$
  $m_0 \leftarrow 0^{|m_1|}$
  $(C, K) \leftarrow \mathsf{KEM.encap}_{\mathrm{pk}_n}()$
  if $(K, h_K) \in \mathsf{H}_{\mathrm{LIST}}$ then
      ABORT
  if $(K, j) \in \mathsf{F}_{\mathrm{LIST}}$ then
      call $\psi_b \leftarrow \mathsf{LR}(j, \vartheta)$
  else
      call $j \leftarrow \mathsf{New}()$
      $\mathsf{F}_{\mathrm{LIST}} \leftarrow \mathsf{F}_{\mathrm{LIST}} \cup \{(K, j)\}$
      call $\psi_b \leftarrow \mathsf{LR}(j, \vartheta)$
  $\mathsf{FL} \leftarrow \mathsf{FL} \cup \{(j, \psi)\}$
  return $(C, \psi_b)$
```

```
$\mathsf{H}(K)$:
  if $(K, h_K) \in \mathsf{H}_{\mathrm{LIST}}$
      return $h_K$
  $h_K \leftarrow \mathsf{PRF}_x(K)$
  $\mathsf{H}_{\mathrm{LIST}} \leftarrow \mathsf{H}_{\mathrm{LIST}} \cup \{(K, h_K)\}$
  return $h_K$

$\mathsf{Dec}(\psi)$:
  call $m \leftarrow \mathsf{Dec}(\psi)$
  return $m$
```

Figure 7: *Description of reduction $\mathcal{C}$ used to prove (10). When $\mathcal{C}$ runs $\mathcal{A}$, it needs to create an environment $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA}}_{\mathsf{Hyb},\,\mathcal{A}}$. It makes* New *queries and specifies the public key index n in its* LR_b *queries. The messages m_0 and m_1 and also C and K are 'created' just as they are in normal* LR_b*, whereas h_K is virtually set to whatever value is used in the game $\mathcal{C}$ itself is playing by $\mathcal{C}$'s calls to* New*,* LR *and* Dec *(from $\mathbf{Exp}^{\mathsf{IND\text{-}PKDM\text{-}CCA}}_{\mathsf{DEM},\,\mathcal{C}}$). Note that $\mathcal{C}$ need not know h_K for this simulation.*

All that remains is bounding the probability of the bad event in games G_2 and G_3, followed by a collection of the various terms into a single bound on the advantage.

The analysis of the bad event in game G_3 is easiest, as here the adversary is given an encryption of a zero string which is clearly not key-dependent (since the adversary directly specifies its length). By simple code-inspection, it emerges that $\mathcal{A}$ can set the flag bad to `true` in two places in G_3: either in a direct oracle query to H on a K that has already been queried to F by LR_0; or if LR_0 calls F on a K that has previously been queried to H directly by $\mathcal{A}$. Intuitively, the former constitutes a break against the one-wayness of the KEM, and the latter should just be very unlikely (although we actually bound it by a break as well to avoid the need for an additional assumption on the way K as output by KEM is distributed). Fig. 8 shows the reduction $\mathcal{B}$ for which

$$\mathbf{Pr}\left[\mathcal{A} \text{ sets bad in } \mathsf{G}_3\right] \leq \mathbf{Adv}^{\mu\mathsf{OW\text{-}CCA}}_{\mathsf{KEM},\,\mathcal{B}}(\lambda) + \mathsf{Coll}_{\mathsf{KEM}}(q_{\mathsf{LR}}, \lambda) \qquad (11)$$

First we observe that if Enc (internally) creates a pair (C, K) and (C', K') satisfying $K = K'$ yet $C \neq C'$ the simulation will with high probability produce $\mathsf{F}(K) \neq \mathsf{F}(K')$, indicating that in that case it is not perfect. However, the event that such a pair is created by a KEM ought to be small. We define $\mathsf{Coll}_{\mathsf{KEM}}(q, \lambda)$ as the probability this happens in q queries to the encapsulation oracle.

In order to simulate correctly, we require that the reductions can make as many New calls as $\mathcal{A}$ can. To do this we can simply set an upper bound on the number of New calls that $\mathcal{A}$ makes, and then restrict the number of calls the reductions can make by this figure.

If a collision as above does not happen then $\mathcal{B}$ creates a perfect simulation of G_3 as long as bad is not set. Moreover, at the very point a query is made that would have caused bad to be set in G_3, the reduction $\mathcal{B}$ uses its KEM-checking oracle KEM.Check to detect that bad was set and retrieves the corresponding key K, plus the index of the Enc query this key belongs to.

As a technical aside, to simulate G_3 the reduction needs to answer the adversary $\mathcal{A}$'s LR_0 queries. Since $\mathcal{A}$ gives out φ and expects an encryption of $0^{|\varphi(\mathbf{sk})|}$, it is necessary (in order to simulate correctly) for $\mathcal{B}$ to learn $|\varphi(\mathbf{sk})|$ without knowing $\mathbf{sk}$. Here the length regularity condition is required: given $\mathbf{pk}$ and φ, we can determine $|\varphi(\mathbf{sk})|$ and thus simulate LR_0.

The analysis of the bad event in G_2 is more problematic and a direct approach (as done for G_3) does not work. Instead, we take inspiration from the "deferred analysis" technique of Gennaro and

```
$\mathcal{B}$ playing $\mathbf{Exp}^{\mu\mathsf{OW\text{-}CCA}}_{\mathsf{KEM},\,\mathcal{B}}(\lambda)$:
  receive pk
  $i \leftarrow 0$
  $\mathsf{H}_{\mathsf{LIST}}, \mathsf{F}_{\mathsf{LIST}}, \mathsf{FL} \leftarrow \emptyset$
  $b' \leftarrow \mathcal{A}^{\mathsf{H},\mathsf{LR},\mathsf{Dec}}(\text{pk})$
  return $\perp$

$\mathsf{H}(K)$:
  if $(K, h_K) \in \mathsf{H}_{\mathsf{LIST}}$
    return $h_K$
  for $(C, h, j) \in \mathsf{F}_{\mathsf{LIST}}$
    if $\mathsf{KEM.Check}(C, K) = \mathtt{true}$
      exit $(j, K)$
  $h_K \xleftarrow{\$} \{0,1\}^\lambda$
  $\mathsf{H}_{\mathsf{LIST}} \leftarrow \mathsf{H}_{\mathsf{LIST}} \cup \{(K, h_K)\}$
  return $h_K$
```

```
$\mathsf{LR}(\varphi, i)$:
  $m_0 \leftarrow 0^{|\varphi(\mathbf{sk})|}$
  call $C \leftarrow \mathsf{Encap}(i)$
  $h_K \leftarrow \mathsf{F}_{\mathsf{SIM}}(C, i)$
  $i \leftarrow i + 1$
  $\psi \leftarrow \mathsf{DEM.Enc}_{h_K}(m_0)$
  $\mathsf{F}_{\mathsf{LIST}} \leftarrow \mathsf{F}_{\mathsf{LIST}} \cup \{(C, \psi, i)\}$
  return $(C, \psi)$

$\mathsf{F}_{\mathsf{SIM}}(C, i)$:
  for $(K, h_K) \in \mathsf{H}_{\mathsf{LIST}}$ do
    if $\mathsf{KEM.Check}(C, K) = \mathtt{true}$ then
      exit $(i, K)$
  if $(C, h_C, j) \in \mathsf{F}_{\mathsf{LIST}}$
    $h \leftarrow h_C$
  else
    $h \xleftarrow{\$} \{0,1\}^\lambda$
  $\mathsf{F}_{\mathsf{LIST}} \leftarrow \mathsf{F}_{\mathsf{LIST}} \cup \{(C, h, i)\}$
  return $h$
```

```
$\mathsf{Dec}(C, \psi, i)$:
  if $(C, \psi, i) \in \mathsf{FL}$ then
    return $\lightning$
  call $K \leftarrow \mathsf{Decap}(C, i)$
  if $K = \perp$ then
    return $\perp_{\mathsf{KEM}}$
  $h_K \leftarrow \mathsf{H}(K)$
  $m \leftarrow \mathsf{DEM.Dec}_{h_K}(\psi)$
  if $m = \perp$ then
    return $\perp_{\mathsf{DEM}}$
  return $m$
```

Figure 8: *Description of reduction $\mathcal{B}$ used to prove (11). When $\mathcal{B}$ runs $\mathcal{A}$, it needs to create an environment $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA}}_{\mathsf{Hyb},\,\mathcal{A}}$. The line "**exit** (j, K)" indicates that $\mathcal{B}$ at that point terminates running $\mathcal{A}$ and returns (j, K) to its own environment (as guess for K_j). As long as* Enc *(internally) does not create a pair (C, K) and (C', K') with $K = K'$ yet $C \neq C'$ the simulation is perfect.*

Shoup [31]. Rather than analysing the bad events in G_2, we will defer the analysis to G_3 (for which we already have a bound). However, it is not at all evident that in the hop G_2 to G_3 the probability the bad flag is set stays the same (as was the case for the deferred analysis by Gennaro and Shoup). Indeed, it is unlikely to be the case, however we are able to show that the difference between the two bad events from occurring is bound by IND-PKDM-CCA advantage of an adversary $\mathcal{D}$ (as described in Fig. 9) against the DEM, or

$$\mathbf{Pr}\left[\mathcal{A} \text{ sets bad in } \mathsf{G}_2\right] - \mathbf{Pr}\left[\mathcal{A} \text{ sets bad in } \mathsf{G}_3\right] \leq \mathbf{Adv}^{\mathsf{IND\text{-}PKDM\text{-}CCA}}_{\mathsf{DEM},\,\mathcal{D}}(\lambda)\,. \tag{12}$$

Similarly to the analysis of (10), it is necessary to translate the function φ into a ϑ, and align the simulated queries correctly. We set this up so that the bad event in the security games corresponds to $\mathcal{D}$ causing an ABORT in the reduction.

1. If $\mathcal{D}$ is in game IND-PKDM-CCA-1 then, unless ABORT occurs, this is a perfect simulation of G_2 for $\mathcal{A}$.

2. If $\mathcal{D}$ is in game IND-PKDM-CCA-0 then, unless ABORT occurs, this is a perfect simulation of G_3 for $\mathcal{A}$.

3. $\mathcal{D}$ will ABORT iff the event bad occurs in (either) G_2 (or G_3).

Consequently we have

$$\begin{aligned} \mathbf{Pr}\left[\mathcal{A} \text{ sets bad in } \mathsf{G}_2\right] &= \mathbf{Pr}\left[\mathcal{D} \text{ sees ABORT in } \mathbf{Exp}^{\mathsf{IND\text{-}PKDM\text{-}CCA\text{-}1}}\right] \\ \mathbf{Pr}\left[\mathcal{A} \text{ sets bad in } \mathsf{G}_3\right] &= \mathbf{Pr}\left[\mathcal{D} \text{ sees ABORT in } \mathbf{Exp}^{\mathsf{IND\text{-}PKDM\text{-}CCA\text{-}0}}\right]. \end{aligned}$$

Since by construction (and definition) we also have

$$\mathbf{Pr}\left[\mathcal{D} \text{ sees ABORT in } \mathbf{Exp}^{\mathsf{IND\text{-}PKDM\text{-}CCA\text{-}}b}\right] = \mathbf{Pr}\left[\mathbf{Exp}^{\mathsf{IND\text{-}PKDM\text{-}CCA\text{-}}b} = 1\right]$$

```
$\mathcal{D}$ playing $\mathbf{Exp}^{\mathsf{IND\text{-}PKDM\text{-}CCA}\text{-}b}_{\mathsf{DEM},\,\mathcal{D}}(\lambda)$:
  $\mathrm{pars} \leftarrow \mathsf{Pg}(1^\lambda)$
  $t \leftarrow 0$
  $\mathbf{sk} \leftarrow ()$
  $\mathsf{H}_{\mathsf{LIST}}, \mathsf{F}_{\mathsf{LIST}}, \mathsf{FL} \leftarrow \emptyset$
  $x \xleftarrow{\$} \{0,1\}^\lambda$
  $b' \leftarrow \mathcal{A}^{\mathsf{New},\mathsf{H},\mathsf{LR}_b,\mathsf{Dec}}(\mathrm{pk})$
  if an ABORT occurs then
      return $1$
  return $0$

$\mathsf{H}(K)$:
  if $(K, h_K) \in \mathsf{H}_{\mathsf{LIST}}$ then
      return $h_K$
  if $(K, *) \in \mathsf{F}_{\mathsf{LIST}}$ then
      ABORT
  $h_K \leftarrow \mathsf{PRF}_x(K)$
  $\mathsf{H}_{\mathsf{LIST}} \leftarrow \mathsf{H}_{\mathsf{LIST}} \cup \{(K, h_K)\}$
  return $h_K$
```

```
$\mathsf{LR}_b(\varphi^{\mathsf{H}}, n)$:
  $\varphi^{\mathsf{H}} \rightarrow \vartheta$
  if $\varphi^{\mathsf{H}}$ makes RO call $K_i$ then
      $h_K \leftarrow \mathsf{PRF}_x(K)$
      $K_\vartheta \leftarrow K_\vartheta \cup K_i$
  $m_1 \leftarrow \vartheta(K_\vartheta)$
  $m_0 \leftarrow 0^{|m_1|}$
  $(C, K) \leftarrow \mathsf{KEM.encap}_{\mathrm{pk}_n}()$
  if $(K, h_K) \in \mathsf{H}_{\mathsf{LIST}}$ then
      ABORT
  if $(K, j) \in \mathsf{F}_{\mathsf{LIST}}$ then
      call $\psi_b \leftarrow \mathsf{LR}(j, \vartheta)$
  else
      call $j \leftarrow \mathsf{New}()$
      $\mathsf{F}_{\mathsf{LIST}} \leftarrow \mathsf{F}_{\mathsf{LIST}} \cup \{(K, j)\}$
      call $\psi_b \leftarrow \mathsf{LR}(j, \vartheta)$
  $\mathsf{FL} \leftarrow \mathsf{FL} \cup \{(j, \psi)\}$
  return $(C, \psi_b)$
```

```
$\mathsf{New}()$:
  $t \leftarrow t + 1$
  $(\mathrm{pk}_t, \mathrm{sk}_t) \leftarrow \mathsf{Kg}(\mathrm{pars})$
  Append $\mathrm{sk}_t$ to $\mathbf{sk}$
  return $\mathrm{pk}_t$

$\mathsf{Dec}(j, \psi)$:
  call $m \leftarrow \mathsf{Dec}(j, \psi)$
  return $m$
```

Figure 9: *Description of reduction $\mathcal{D}$ used to prove (12). When $\mathcal{D}$ runs $\mathcal{A}$, it needs to create an environment $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA}}_{\mathsf{Hyb},\,\mathcal{A}}$. The messages m_0 and m_1 and also C and K are 'created' just as they are in normal LR_b, whereas h_K is virtually set to whatever value is used in the game $\mathcal{D}$ itself is playing by $\mathcal{D}$'s calls to* New, LR *and* Dec *(from $\mathbf{Exp}^{\mathsf{IND\text{-}PKDM\text{-}CCA}}_{\mathsf{DEM},\,\mathcal{D}}$). The number of keypairs $\mathcal{D}$ can ask for is upper-bounded by the number of* New *queries $\mathcal{A}$ makes. Note that $\mathcal{D}$ need not know h_K for this simulation.*

and so our claim (12) follows. Finally we put all of the above together and arrive at the claimed bound .

$$
\begin{aligned}
\mathbf{Adv}^{\mathsf{IND\text{-}KDM\text{-}CCA}[\Phi]}_{\mathsf{Hyb},\,\mathcal{A}}(\lambda) &= \left|\mathsf{Pr}[\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA\text{-}1}}_{\mathsf{Hyb},\,\mathcal{A}}(\lambda) = 1] - \mathbf{Pr}\left[\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA\text{-}0}}_{\mathsf{Hyb},\,\mathcal{A}}(\lambda) = 1\right]\right| \\
&= \left|\mathbf{Pr}\left[\mathsf{G_0}^{\mathcal{A}} = 1\right] - \mathbf{Pr}\left[\mathsf{G_5}^{\mathcal{A}} = 1\right]\right| && \text{[by (2),(3)]} \\
&= \Big|\mathbf{Pr}\left[\mathsf{G_0}^{\mathcal{A}} = 1\right] - \mathbf{Pr}\left[\mathsf{G_1}^{\mathcal{A}} = 1\right] + \mathbf{Pr}\left[\mathsf{G_1}^{\mathcal{A}} = 1\right] - \mathbf{Pr}\left[\mathsf{G_2}^{\mathcal{A}} = 1\right] + \mathbf{Pr}\left[\mathsf{G_2}^{\mathcal{A}} = 1\right] \\
&\quad \text{-}\, \mathbf{Pr}\left[\mathsf{G_3}^{\mathcal{A}} = 1\right] + \mathbf{Pr}\left[\mathsf{G_3}^{\mathcal{A}} = 1\right] - \mathbf{Pr}\left[\mathsf{G_4}^{\mathcal{A}} = 1\right] + \mathbf{Pr}\left[\mathsf{G_4}^{\mathcal{A}} = 1\right] - \mathbf{Pr}\left[\mathsf{G_5}^{\mathcal{A}} = 1\right]\Big| \\
&\le \Big|\mathbf{Pr}\left[\mathsf{G_0}^{\mathcal{A}} = 1\right] - \mathbf{Pr}\left[\mathsf{G_1}^{\mathcal{A}} = 1\right] + \mathbf{Adv}^{\mathsf{PRF}}_{F,\,\mathcal{A}}(\lambda) + \mathbf{Pr}\left[\mathsf{G_2}^{\mathcal{A}} = 1\right] - \mathbf{Pr}\left[\mathsf{G_3}^{\mathcal{A}} = 1\right] \\
&\qquad + \mathbf{Adv}^{\mathsf{PRF}}_{F,\,\mathcal{A}}(\lambda) + \mathbf{Pr}\left[\mathsf{G_4}^{\mathcal{A}} = 1\right] - \mathbf{Pr}\left[\mathsf{G_5}^{\mathcal{A}} = 1\right]\Big| && \text{[by (6), (8)]} \\
&\le \Big|\mathbf{Pr}\left[\mathcal{A} \text{ sets bad in } \mathsf{G_1}\right]\Big| \qquad + \mathbf{Pr}\left[\mathcal{A} \text{ sets bad in } \mathsf{G_4}\right] + && \text{[by (4), (5)]} \\
&\qquad + \mathbf{Adv}^{\mathsf{IND\text{-}PKDM\text{-}CCA}}_{\mathsf{DEM},\,\mathcal{C}}(\lambda) + 2\mathbf{Adv}^{\mathsf{PRF}}_{F,\,\mathcal{A}}(\lambda) && \text{[by (10)]} \\
&\le \Big|\mathbf{Pr}\left[\mathcal{A} \text{ sets bad in } \mathsf{G_2}\right] + \mathbf{Adv}^{\mathsf{PRF}}_{F,\,\mathcal{A}}(\lambda)\Big| + \mathbf{Pr}\left[\mathcal{A} \text{ sets bad in } \mathsf{G_3}\right] + \mathbf{Adv}^{\mathsf{PRF}}_{F,\,\mathcal{A}}(\lambda) + && \text{[by (7), (9)]} \\
&\qquad + \mathbf{Adv}^{\mathsf{IND\text{-}PKDM\text{-}CCA}}_{\mathsf{DEM},\,\mathcal{C}}(\lambda) + 2\mathbf{Adv}^{\mathsf{PRF}}_{F,\,\mathcal{A}}(\lambda) \\
&\le 2\mathbf{Pr}\left[\mathcal{A} \text{ sets bad in } \mathsf{G_3}\right] + \mathbf{Adv}^{\mathsf{IND\text{-}PKDM\text{-}CCA}}_{\mathsf{DEM},\,\mathcal{C}}(\lambda) + \mathbf{Adv}^{\mathsf{IND\text{-}PKDM\text{-}CCA}}_{\mathsf{DEM},\,\mathcal{D}}(\lambda) + 4\mathbf{Adv}^{\mathsf{PRF}}_{F,\,\mathcal{A}}(\lambda) && \text{[by (12)]} \\
&\le 2\mathbf{Adv}^{\mathsf{IND\text{-}PKDM\text{-}CCA}}_{\mathsf{DEM},\,\mathcal{C}}(\lambda) + 2\mathbf{Adv}^{\mu\mathsf{OW\text{-}CCA}}_{\mathsf{KEM},\,\mathcal{B}}(\lambda) + 2\mathsf{Coll}_{\mathsf{KEM}}(q_{\mathsf{LR}}, \lambda) + 4\mathbf{Adv}^{\mathsf{PRF}}_{F,\,\mathcal{A}}(\lambda)\,. && \text{[by (11)]}
\end{aligned}
$$

□

4 Separation Results

In this section, we detail two schemes that are secure under standard security definitions, yet are insecure when the adversary has access to a key cycle of length 2. We recall that security against 2-cycles is referred to as CYC-security. We cast the two schemes, both reliant on the SXDH assumption [8] and containing somewhat 'artificial properties,' in the hybrid framework. Since there is no need to make anything other than cosmetic changes to the schemes to make them hybrid, the security and insecurity results relating the full schemes follow from the original papers (and we refer to the original papers for full details and proofs).

By inspection, it also easy to see that both KEMs are $\mathsf{IND\text{-}CPA}$ secure under the SXDH assumption and that the DEMs are one-time $\mathsf{IND\text{-}CPA}$ secure, assuming a key derivation function is pseudorandom. The reason the schemes nonetheless fail to fall under our framework, is that the key derivation function is only applied to *part* of the key. Indeed, it is the other part of the key (and how it is used by the DEM) that has been carefully crafted by the original authors to ensure CYC-insecurity. As a result, these existing counterexamples also serve as proof that the KEM–DEM framework *per se* does not provide any 'leverage' when it comes to increasing resistance against key-dependent message attacks.

4.1 Writing Acar et al.'s es_2 as a KEM and DEM

Acar et al. [3] present schemes that are $\mathsf{IND\text{-}R}$-secure (indistinguishable from random) if the SXDH problem is hard, yet not CYC-secure. Let GS be a group scheme for which SXDH is hard. The original paper shows $\mathsf{es}_2 = (\mathsf{es}_2.\mathsf{Pg}, \mathsf{es}_2.\mathsf{Kg}, \mathsf{es}_2.\mathsf{Enc}, \mathsf{es}_2.\mathsf{Dec}, \mathsf{es}_2.\mathsf{MsgR}, \mathsf{es}_2.\mathsf{CtxtR})$, an asymmetric scheme. The paper also presents a symmetric scheme but our focus will be on es_2. The decryption keys are in the message space.

Lemmas 3 and 4 of [3] prove this scheme to be $\mathsf{IND\text{-}R}$-secure but not CYC-secure. We will now cast this scheme in a hybrid framework and comment on the security properties. Fig. 10 details hybrid scheme $\mathsf{Hyb}_1 = (\mathsf{Hyb}_1.\mathsf{Pg}, \mathsf{Hyb}_1.\mathsf{Kg}, \mathsf{Hyb}_1.\mathsf{KEM.encap}, \mathsf{Hyb}_1.\mathsf{DEM.Enc}, .\mathsf{Hyb}_1.\mathsf{KEM.decap}, \mathsf{Hyb}_1.\mathsf{DEM.Dec})$ which is a straightforward adaptation of the es_2 scheme. The KEM part generates keys that are uniform over the keyspace, however it is important to note that the keyspace of the DEM part depends on pars, so we must allow some joint parameter generation between the KEM and the DEM. The changes from the original scheme are only cosmetic, and as a result the security properties of the original scheme, namely $\mathsf{IND\text{-}R}$-secure yet KDM- and CYC-insecure still hold.

The KEM as described by us is $\mathsf{IND\text{-}CPA}$ secure under the SXDH assumption: parameters, public key, key encapsulation and encapsulated key form two independent DDH instances in each of the groups, so replacing any of these (in particular the keys) with unrelated random elements from the same group will go unnoticed (assuming SXDH). The DEM is deterministic, and hence it can clearly not be multi-time secure. However, it is one-time secure: For every T_1 and m_2, there is a unique Ω_1 such that $T_1 = \Omega_1^{m_2}$, indicating that m_2 is information-theoretically hidden given T_1 and similarly for m_1 and T_2; moreover, if H is a balanced function, then $m_1 + H(\ldots, Z_1)$ operates as a perfect one-time pad. Thus the DEM is perfectly one-time secure if H is a balanced function. If H is a pseudorandom generator, security degrades to one-time $\mathsf{IND\text{-}CPA}$ security.

As an aside, while Acar et al. noted the scheme was insecure against 2-cycles, they did not remark on 1-cycles. It is easy to see that security also breaks down in this case: If $\varphi(\mathrm{sk}) = \varphi(x_1, x_2) = (x_1, x_2)$ then an adversary can disinguish the encryption of $(m_1, m_2) = (x_1, x_2)$: $T_1 = g_1^{x_1 u_1/x_2}$ and $T_2 = g_2^{x_2 u_2/x_1}$ so $\mathbf{e}(T_1, T_2) = \mathbf{e}(g_1, g_2)^{x_1 u_1 x_2 u_2/x_1 x_2} = \mathbf{e}(g_1, g_2)^{u_1 u_2} = \mathbf{e}(U_1, U_2)$ so the adversary can simply check if these values coincide.

```
$\underline{\mathsf{Hyb}_1.\mathsf{Pg}(1^\lambda)}$
$(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbf{e}, g_1, g_2) \xleftarrow{\$} \mathsf{GS}(1^\lambda)$
$\mathrm{pars} \leftarrow (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbf{e}, g_1, g_2)$
return pars

$\underline{\mathsf{Hyb}_1.\mathsf{Kg}(\mathrm{pars})}$
$x_1, x_2 \xleftarrow{\$} \mathbb{Z}_p^*$
$X_1 \leftarrow g_1^{x_1}$ ; $X_2 \leftarrow g_2^{x_2}$
$\mathrm{sk} \leftarrow (x_1, x_2)$ ; $\mathrm{pk} \leftarrow (X_1, X_2)$
return $(\mathrm{pk}, \mathrm{sk})$

$\underline{\mathsf{Hyb}_1.\mathsf{KEM.encap}(\mathrm{pars}, \mathrm{pk})}$
$(X_1, X_2) \leftarrow \mathrm{pk}$ ; $y_1, y_2, u_1, u_2 \xleftarrow{\$} \mathbb{Z}_p^*$
$Y_1 \leftarrow g_1^{y_1}$ ; $U_1 \leftarrow g_1^{u_1}$ ; $Z_1 \leftarrow X_1^{y_1}$
$Y_2 \leftarrow g_2^{y_2}$ ; $U_2 \leftarrow g_2^{u_2}$ ; $Z_2 \leftarrow X_2^{y_2}$
$\Omega_1 \leftarrow X_1^{u_1}$ ; $\Omega_2 \leftarrow X_2^{u_2}$
$C \leftarrow (Y_1, Y_2, U_1, U_2)$
$K \leftarrow (Z_1, Z_2, \Omega_1, \Omega_2)$
return $(C, K)$
```

```
$\underline{\mathsf{Hyb}_1.\mathsf{DEM.Enc}(\mathrm{pars}, \mathrm{pk}, K, (m_1, m_2))}$
$(Z_1, Z_2, \Omega_1, \Omega_2) \leftarrow K$
$T_1 \leftarrow \Omega_1^{1/m_2}$ ; $T_2 \leftarrow \Omega_2^{1/m_1}$
$c_1 \leftarrow m_1 + H(\mathrm{pars}, 1, Z_1)$
$c_2 \leftarrow m_2 + H(\mathrm{pars}, 2, Z_2)$
$\psi \leftarrow (T_1, T_2, c_1, c_2)$
return $\psi$

$\underline{\mathsf{Hyb}_1.\mathsf{KEM.decap}(\mathrm{pars}, \mathrm{sk}, C)}$
$(x_1, x_2) \leftarrow \mathrm{sk}$ ; $(Y_1, Y_2, U_1, U_2) \leftarrow C$
$Z_1 \leftarrow Y_1^{x_1}$ ; $Z_2 \leftarrow Y_2^{x_2}$
$\Omega_1 \leftarrow U_1^{x_1}$ ; $\Omega_2 \leftarrow U_2^{x_2}$
$K \leftarrow (Z_1, Z_2, \Omega_1, \Omega_2)$
return $K$

$\underline{\mathsf{Hyb}_1.\mathsf{DEM.Dec}(\mathrm{pars}, K, \psi)}$
$(T_1, T_2, c_1, c_2) \leftarrow \psi$
$(Z_1, Z_2, \Omega_1, \Omega_2) \leftarrow K$
$m_1 \leftarrow c_1 - H(\mathrm{pars}, 1, Z_1)$
$m_2 \leftarrow c_2 - H(\mathrm{pars}, 2, Z_2)$
return $(m_1, m_2)$
```

Figure 10: *Hybrid scheme* Hyb_1 *based on* es_2 *of [3]. KEM is* IND-CPA *if DDH in* $\mathbb{G}_1$ *holds.*

4.2 Writing Cash et al.'s Π_{CPA} as a KEM and DEM

The encryption scheme of Cash et al. [26] $\Pi_{\mathsf{CPA}} = (\Pi_{\mathsf{CPA}}.\mathsf{Pg}, \Pi_{\mathsf{CPA}}.\mathsf{Kg}, \Pi_{\mathsf{CPA}}.\mathsf{Enc}, \Pi_{\mathsf{CPA}}.\mathsf{Dec})$, also uses asymmetric bilinear groups $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ of prime order p, and assumes that $\mathbb{G}_1$ and $\mathbb{G}_2$ are distinct and that the DDH assumption holds in both (aka the SXDH assumption). The scheme includes functions encode : $\mathcal{M} \to \{0,1\}^{l(\lambda)}$ and decode : $\{0,1\}^{l(\lambda)} \to \mathcal{M}$, which denote an invertible encoding scheme, where $l(\lambda)$ is the polynomial length of the encoded message. Let $F : \mathbb{G}_T \to \{0,1\}^{l(\lambda)}$ be a pseudorandom generator.

The authors show that this scheme is IND-CPA secure, however when given a circular encryption of two keys, an adversary can distinguish another ciphertext with probability $1/2$. In fact, with probability 1/2 over the coins used in key generation, the adversary can recover both secret keys. In the appendix of the full version of their paper, the authors give another scheme that is IND-CPA secure without using the 'group-switching' technique, and experiences catastrophic collapse in the presence of a 2-cycle, with even higher adversarial success probability.

Fig. 11 details hybrid scheme $\mathsf{Hyb}_2 = (\mathsf{Hyb}_2.\mathsf{Pg}, \mathsf{Hyb}_2.\mathsf{Kg}, \mathsf{Hyb}_2.\mathsf{KEM.encap}, \mathsf{Hyb}_2.\mathsf{DEM.Enc}, \mathsf{Hyb}_2.\mathsf{KEM.decap}, \mathsf{Hyb}_2.\mathsf{DEM.Dec})$ which is a straightforward casting of the Π_{CPA} scheme. in the hybrid framework. Let us see what we can say about the relevant security properties. Again, the changes are only cosmetic, the (in)security properties of the original scheme, namely IND-CPA-secure yet CYC-insecure carry over without reserve. Here we consider the security of the KEM and the DEM as defined by us. The DEM is secure for arguments similar to those used for the Acar et al. scheme: it is one-time IND-CPA secure provided that F is a pseudorandom generator.

By construction, if $\beta = 0$ the KEM part takes place primarily in $\mathbb{G}_1$, whereas $\beta = 1$ there is a symmetric move to $\mathbb{G}_2$. Consequently, we only need to analyse the KEM security for $\beta = 0$ and the $\beta = 1$ follows by symmetry. For $\beta = 0$, we have that (g_1, C_1, Y_2, K_2) forms a DDH tuple; if we substitute $\vec{Y_1} = g_1^{x_1}$ for Y_1 and inverse R through the pairing resulting in $\vec{R} = \mathbf{e}(R, g_2)$ and let $\tilde{C_2} = \vec{R}.\vec{Y_1}^{r}$ (all this only makes the adversary's life easier) then the same holds for the tuple $(g_1, \vec{Y_1}, \vec{R}, \tilde{C_2}/\vec{R})$. We conclude that the KEM (for $\beta = 0$) is one-way secure if the CDH assumption holds in $\mathbb{G}_1$ and and IND-CPA if the DDH assumption holds (in $\mathbb{G}_1$).

We observe that the scheme Π_{CPA} is not trivially insecure under 1-cycles. In fact, if the square decision Diffie Hellman (SDDH) [11] assumption holds in both groups the scheme seems 1-cycle secure and we conjecture that the scheme is actually fully single-key $\mathsf{IND\text{-}KDM}[\Phi]$ secure in the Generic Group Model [45] (adapted to the asymmetric pairing setting).

```
$\mathsf{Hyb}_2.\mathsf{Pg}(1^\lambda)$
$(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbf{e}, g_1, g_2) \xleftarrow{\$} \mathsf{GS}(1^\lambda)$
$\mathrm{pars} \leftarrow (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbf{e}, g_1, g_2)$
return $\mathrm{pars}$

$\mathsf{Hyb}_2.\mathsf{Kg}(\mathrm{pars})$
$x_1, x_2 \xleftarrow{\$} \mathbb{Z}_p^*$ ; $\beta \xleftarrow{\$} \{0,1\}$
$Y_1 \leftarrow \mathbf{e}(g_1, g_2)^{x_1}$
if $\beta = 0$:
  $Y_2 \leftarrow g_1^{x_2}$
if $\beta = 1$:
  $Y_2 \leftarrow g_2^{x_2}$
$\mathrm{sk} \leftarrow (\beta, x_1, x_2)$ ; $\mathrm{pk} \leftarrow (\beta, Y_1, Y_2)$
return $(\mathrm{pk}, \mathrm{sk})$

$\mathsf{Hyb}_2.\mathsf{KEM.encap}(\mathrm{pars}, \mathrm{pk})$
$(\beta, Y_1, Y_2) \leftarrow \mathrm{pk}$
$r \xleftarrow{\$} \mathbb{Z}_p^*$ ; $R \xleftarrow{\$} \mathbb{G}_T$
$C_2 \leftarrow R.Y_1^r$ ; $K_1 \leftarrow R$ ; $K_2 \leftarrow Y_2^r$
if $\beta = 0$:
  $C_1 \leftarrow g_1^r$
if $\beta = 1$:
  $C_1 \leftarrow g_2^r$
$C \leftarrow (C_1, C_2)$ ; $K \leftarrow (K_1, K_2)$
return $(C, K)$
```

```
$\mathsf{Hyb}_2.\mathsf{DEM.Enc}(\mathrm{pars}, \mathrm{pk}, K, M)$
$(\beta, Y_1, Y_2) \leftarrow \mathrm{pk}$ ; $(\alpha, m_1, m_2) \leftarrow M$
$(K_1, K_2) \leftarrow K$
$I \leftarrow F(K_1) \oplus \mathsf{encode}(M)$
$C_4 \leftarrow I$
if $\beta = 0$:
  $C_3 \leftarrow K_2^{m_2}.g_1^{m_2}$
if $\beta = 1$:
  $C_3 \leftarrow K_2^{m_2}$
$\psi \leftarrow (C_3, C_4)$
return $\psi$

$\mathsf{Hyb}_2.\mathsf{KEM.decap}(\mathrm{pars}, \mathrm{sk}, C)$
$(\beta, x_1, x_2) \leftarrow \mathrm{sk}$ ; $(C_1, C_2) \leftarrow C$
if $\beta = 0$:
  $R \leftarrow (C_2/\mathbf{e}(C_1, g_2)^{x_1})$ ; $K_2 \leftarrow C_1^{x_2}$
if $\beta = 1$:
  $R \leftarrow (C_2/\mathbf{e}(g_1, C_1)^{x_1})$ ; $K_2 \leftarrow C_1^{x_2}$
$K_1 \leftarrow R$ ; $K \leftarrow (K_1, K_2)$
return $K$

$\mathsf{Hyb}_2.\mathsf{DEM.Dec}(\mathrm{pars}, K, \psi)$
$(C_3, C_4) \leftarrow \psi$ ; $(K_1, K_2) \leftarrow K$
$M' \leftarrow F(K_1) \oplus C_4$
$M \leftarrow \mathsf{decode}(M')$
return $M$
```

Figure 11: *Asymmetric scheme* Hyb_2 *based on* Π_{CPA} *of Cash et al. [26]. Message space is* $\mathcal{M} = \{0,1\} \times \mathbb{Z}_p^* \times \mathbb{Z}_p^*$ *thus* m_1 *and* m_2 *must be non-zero, these values can be included in the message space by proper encoding. The ciphertext space is* $\mathbb{G}_{\beta+1} \times \mathbb{G}_T \times \mathbb{G}_{\beta+1} \times \{0,1\}^{l(\lambda)}$*. Note that* Y_2 *may be either in* $\mathbb{G}_1$ *or* $\mathbb{G}_2$ *depending on the structure of the public key.*

5 Conclusions and Open Problems

As stated, our result is very general as it incorporates active attacks, and allows KDM functions that call the random oracle. The proof method incorporates the use of a PRF and the non-standard $\mathsf{IND\text{-}PKDM\text{-}CCA}$ notion of security on the DEM (notably, a notion equivalent to $\mathsf{IND\text{-}CCA}$). In the scenario where the adversary's KDM functions *cannot* call the random oracle, direct reductions to $\mathsf{IND\text{-}CCA}$ security of the DEM are possible, however there is an additional bad event caused when there exists (C, i) and (C', i') such that $\mathsf{KEM.decap}_{\mathrm{sk}_i}(C) = \mathsf{KEM.decap}_{\mathrm{sk}_{i'}}(C') = K$ and the adversary manages to decapsulate to a previously seen protokey, without triggering the forbidden list FL. It is consequently possible to bound both of the bad events together.

Once we move to the scenario where functions can call the random oracle, it is necessary to consider how the adversary could exploit this issue. First of all the adversary makes an arbitrary LR query, receiving an encryption under some protokey K_1 encapsulated in C_1. It then makes another LR query, this time submitting a function that depends on the K_1 used in its previous query, receiving (C_2, ψ_2). Then a decryption query of the form $m' \leftarrow \mathsf{Dec}(C_1, \psi_2)$ does not fall on the forbidden list and could yield information about m' to the attacker. From this perspective it is a challenge to negotiate the issue of simulating the keys in the DEM hop without using a PRF, and to realise direct reductions to a standard security notion such as IND-CCA.

The natural open problem that arises from this work is consideration of KDM security of the hybrid framework in the standard model. The main goal is to develop a composition theorem, with or without a KDF, that states standard, individual security notions for the KEM and the DEM to yield a KDM secure hybrid system, with security against passive and active attacks.

Acknowledgments

We would like to thank Pooya Farshim for his contribution to initial discussions, and Bogdan Warinschi for ideas and input throughout. In addition, we would like to thank the anonymous conference reviewers for their comments and suggestions.

References

[1] Abadi, M., Rogaway, P.: Reconciling two views of Cryptography (the Computational Soundness of Formal Encryption). J. Cryptology 15(2), 103–127 (2002) Cited on page 3.

[2] Abe, M., Gennaro, R., Kurosawa, K., Shoup, V.: Tag-KEM/DEM: A new Framework for Hybrid Encryption and a new Analysis of Kurosawa-Desmedt KEM. In: Cramer, R. (ed.) EUROCRYPT. Lecture Notes in Computer Science, vol. 3494, pp. 128–146. Springer (2005) Cited on pages 3 and 6.

[3] Acar, T., Belenkiy, M., Bellare, M., Cash, D.: Cryptographic Agility and its Relation to Circular Encryption. In: Gilbert [33], pp. 403–422 Cited on pages 3, 4, 5, 18, and 19.

[4] Adão, P., Bana, G., Herzog, J., Scedrov, A.: Soundness of Formal Encryption in the Presence of Key-Cycles. In: di Vimercati, S.D.C., Syverson, P.F., Gollmann, D. (eds.) ESORICS. Lecture Notes in Computer Science, vol. 3679, pp. 374–396. Springer (2005) Cited on pages 3 and 5.

[5] Alperin-Sheriff, J., Peikert, C.: Circular and KDM Security for Identity-Based Encryption. In: Fischlin et al. [30], pp. 334–352 Cited on page 5.

[6] Applebaum, B.: Key-Dependent Message Security: Generic Amplification and Completeness. In: Paterson [42], pp. 527–546 Cited on page 5.

[7] Applebaum, B., Cash, D., Peikert, C., Sahai, A.: Fast Cryptographic Primitives and Circular-Secure Encryption based on Hard Learning Problems. In: Halevi, S. (ed.) CRYPTO. Lecture Notes in Computer Science, vol. 5677, pp. 595–618. Springer (2009) Cited on page 5.

[8] Ateniese, G., Camenisch, J., Hohenberger, S., de Medeiros, B.: Practical group signatures without random oracles. IACR Cryptology ePrint Archive 2005, 385 (2005) Cited on page 17.

[9] Backes, M., Dürmuth, M., Unruh, D.: OAEP is secure under Key-Dependent Messages. In: Pieprzyk, J. (ed.) ASIACRYPT. Lecture Notes in Computer Science, vol. 5350, pp. 506–523. Springer (2008) Cited on pages 5, 8, and 9.

[10] Backes, M., Pfitzmann, B., Scedrov, A.: Key-Dependent Message Security under active attacks - BRSIM/UC-Soundness of Symbolic Encryption with Key Cycles. In: CSF. pp. 112–124. IEEE Computer Society (2007) Cited on pages 5 and 8.

[11] Bao, F., Deng, R.H., Zhu, H.: Variations of Diffie-Hellman Problem. In: Qing, S., Gollmann, D., Zhou, J. (eds.) ICICS. Lecture Notes in Computer Science, vol. 2836, pp. 301–312. Springer (2003) Cited on page 20.

[12] Barak, B., Haitner, I., Hofheinz, D., Ishai, Y.: Bounded Key-Dependent Message Security. In: Gilbert [33], pp. 423–444 Cited on page 5.

[13] Bellare, M., Cash, D., Keelveedhi, S.: Ciphers that securely encipher their own keys. In: Chen, Y., Danezis, G., Shmatikov, V. (eds.) ACM Conference on Computer and Communications Security. pp. 423–432. ACM (2011) Cited on page 5.

[14] Bellare, M., Desai, A., Pointcheval, D., Rogaway, P.: Relations among notions of security for Public-Key Encryption schemes. In: Krawczyk, H. (ed.) CRYPTO. Lecture Notes in Computer Science, vol. 1462, pp. 26–45. Springer (1998) Cited on pages 3 and 6.

[15] Bellare, M., Keelveedhi, S.: Authenticated and Misuse-Resistant Encryption of Key-Dependent Data. In: Rogaway [43], pp. 610–629 Cited on page 5.

[16] Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Denning, D.E., Pyle, R., Ganesan, R., Sandhu, R.S., Ashby, V. (eds.) ACM Conference on Computer and Communications Security. pp. 62–73. ACM (1993) Cited on page 3.

[17] Bellare, M., Rogaway, P.: The security of triple encryption and a framework for code-based game-playing proofs. In: Vaudenay, S. (ed.) EUROCRYPT. Lecture Notes in Computer Science, vol. 4004, pp. 409–426. Springer (2006) Cited on pages 5 and 9.

[18] Black, J., Rogaway, P., Shrimpton, T.: Encryption-scheme security in the presence of Key-Dependent Messages. In: Nyberg, K., Heys, H.M. (eds.) Selected Areas in Cryptography. Lecture Notes in Computer Science, vol. 2595, pp. 62–75. Springer (2002) Cited on pages 3, 7, 9, and 11.

[19] Boneh, D., Halevi, S., Hamburg, M., Ostrovsky, R.: Circular-Secure Encryption from Decision Diffie-Hellman. In: Wagner, D. (ed.) CRYPTO. Lecture Notes in Computer Science, vol. 5157, pp. 108–125. Springer (2008) Cited on pages 3 and 4.

[20] Brakerski, Z., Goldwasser, S.: Circular and Leakage Resilient Public-Key Encryption under Subgroup Indistinguishability - (or: Quadratic Residuosity strikes back). In: Rabin, T. (ed.) CRYPTO. Lecture Notes in Computer Science, vol. 6223, pp. 1–20. Springer (2010) Cited on page 5.

[21] Brakerski, Z., Goldwasser, S., Kalai, Y.T.: Black-box circular-secure encryption beyond affine functions. In: Ishai, Y. (ed.) TCC. Lecture Notes in Computer Science, vol. 6597, pp. 201–218. Springer (2011) Cited on page 5.

[22] Brakerski, Z., Vaikuntanathan, V.: Fully Homomorphic Encryption from Ring-LWE and security for Key Dependent Messages. In: Rogaway [43], pp. 505–524 Cited on page 5.

[23] Camenisch, J., Chandran, N., Shoup, V.: A Public Key Encryption scheme secure against Key Dependent chosen plaintext and adaptive chosen ciphertext attacks. In: Joux, A. (ed.) EUROCRYPT. Lecture Notes in Computer Science, vol. 5479, pp. 351–368. Springer (2009) Cited on pages 5 and 7.

[24] Camenisch, J., Lysyanskaya, A.: An efficient system for non-transferable Anonymous Credentials with optional anonymity revocation. In: Pfitzmann, B. (ed.) EUROCRYPT. Lecture Notes in Computer Science, vol. 2045, pp. 93–118. Springer (2001) Cited on page 3.

[25] Canetti, R., Kalai, Y.T., Varia, M., Wichs, D.: On symmetric encryption and Point Obfuscation. In: Micciancio, D. (ed.) TCC. Lecture Notes in Computer Science, vol. 5978, pp. 52–71. Springer (2010) Cited on page 5.

[26] Cash, D., Green, M., Hohenberger, S.: New definitions and separations for Circular Security. In: Fischlin et al. [30], pp. 540–557 Cited on pages 3, 4, 5, 19, and 20.

[27] Cramer, R., Shoup, V.: Design and analysis of practical Public-Key Encryption schemes secure against adaptive chosen ciphertext attack. IACR Cryptology ePrint Archive 2001, 108 (2001) Cited on pages 3 and 6.

[28] Dent, A.W.: A designer's guide to KEMs. In: Paterson, K.G. (ed.) IMA Int. Conf. Lecture Notes in Computer Science, vol. 2898, pp. 133–151. Springer (2003) Cited on pages 4, 6, 9, and 11.

[29] Dent, A.W.: On the Equivalence of two models for Key-Dependent-Message Encryption. IACR Cryptology ePrint Archive 2009, 572 (2009) Cited on page 8.

[30] Fischlin, M., Buchmann, J., Manulis, M. (eds.): Public Key Cryptography - PKC 2012 - 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, May 21-23, 2012. Proceedings, Lecture Notes in Computer Science, vol. 7293. Springer (2012) Cited on pages 21 and 22.

[31] Gennaro, R., Shoup, V.: A note on an encryption scheme of Kurosawa and Desmedt. IACR Cryptology ePrint Archive 2004, 194 (2004) Cited on pages 4 and 16.

[32] Gentry, C.: Fully Homomorphic Encryption using ideal lattices. In: Mitzenmacher, M. (ed.) STOC. pp. 169–178. ACM (2009) Cited on page 5.

[33] Gilbert, H. (ed.): Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings, Lecture Notes in Computer Science, vol. 6110. Springer (2010) Cited on page 21.

[34] Haitner, I., Holenstein, T.: On the (im)possibility of Key Dependent Encryption. In: Reingold, O. (ed.) TCC. Lecture Notes in Computer Science, vol. 5444, pp. 202–219. Springer (2009) Cited on page 5.

[35] Halevi, S., Krawczyk, H.: Security under Key-Dependent Inputs. In: Ning, P., di Vimercati, S.D.C., Syverson, P.F. (eds.) ACM Conference on Computer and Communications Security. pp. 466–475. ACM (2007) Cited on page 5.

[36] Hofheinz, D., Kiltz, E.: Secure hybrid encryption from weakened key encapsulation. In: Menezes, A. (ed.) CRYPTO. Lecture Notes in Computer Science, vol. 4622, pp. 553–571. Springer (2007) Cited on pages 3 and 6.

[37] Hofheinz, D., Unruh, D.: Towards Key-Dependent Message security in the Standard Model. In: Smart, N.P. (ed.) EUROCRYPT. Lecture Notes in Computer Science, vol. 4965, pp. 108–126. Springer (2008) Cited on page 5.

[38] Kremer, S., Steel, G., Warinschi, B.: Security for Key Management Interfaces. In: CSF. pp. 266–280. IEEE Computer Society (2011) Cited on page 8.

[39] Kurosawa, K., Desmedt, Y.: A new Paradigm of Hybrid Encryption scheme. In: Franklin, M.K. (ed.) CRYPTO. Lecture Notes in Computer Science, vol. 3152, pp. 426–442. Springer (2004) Cited on pages 3 and 6.

[40] Laud, P., Corin, R.: Sound computational interpretation of formal encryption with composed keys. In: Lim, J.I., Lee, D.H. (eds.) ICISC. Lecture Notes in Computer Science, vol. 2971, pp. 55–66. Springer (2003) Cited on page 5.

[41] Malkin, T., Teranishi, I., Yung, M.: Efficient circuit-size independent Public Key Encryption with KDM Security. In: Paterson [42], pp. 507–526 Cited on page 5.

[42] Paterson, K.G. (ed.): Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings, Lecture Notes in Computer Science, vol. 6632. Springer (2011) Cited on pages 21 and 23.

[43] Rogaway, P. (ed.): Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings, Lecture Notes in Computer Science, vol. 6841. Springer (2011) Cited on page 22.

[44] Rothblum, R.: On the Circular Security of Bit-Encryption. IACR Cryptology ePrint Archive 2012, 102 (2012) Cited on page 5.

[45] Shoup, V.: Lower bounds for Discrete Logarithms and related problems. In: Fumy, W. (ed.) EUROCRYPT. Lecture Notes in Computer Science, vol. 1233, pp. 256–266. Springer (1997) Cited on page 20.

[46] Shoup, V.: Sequences of games: a tool for taming complexity in security proofs. IACR Cryptology ePrint Archive 2004, 332 (2004) Cited on pages 5 and 9.

[47] Unruh, D.: Programmable encryption and Key-Dependent Messages. IACR Cryptology ePrint Archive 2012, 423 (2012) Cited on pages 5, 7, and 8.

A Standard Security Notions for KEMs and DEMs

Now we look at the security definitions for the individual components of the hybrid system, and their security definitions.

Definition A.1. Let $\mathsf{KEM} = (\mathsf{KEM.Pg}, \mathsf{KEM.Kg}, \mathsf{KEM.encap}, \mathsf{KEM.decap})$ be a key encapsulation mechanism. Then the μOW-CCA advantage of an adversary $\mathcal{A}$ against KEM is defined by

$$\mathbf{Adv}^{\mu\mathsf{OW\text{-}CCA}}_{\mathsf{KEM},\,\mathcal{A}}(\lambda) \overset{def}{=} \mathbf{Pr}\left[\mathbf{Exp}^{\mu\mathsf{OW\text{-}CCA}}_{\mathsf{KEM},\,\mathcal{A}}(\lambda) = 1\right]$$

where the experiment $\mathbf{Exp}^{\mu\mathsf{OW\text{-}CCA}}_{\mathsf{KEM},\,\mathcal{A}}(\lambda)$ is given in Fig. 12.

```
$\mathbf{Exp}^{\mu\mathsf{OW\text{-}CCA}}_{\mathsf{KEM},\,\mathcal{A}}(\lambda)$:
  pars $\leftarrow \mathsf{KEM.Pg}(1^\lambda)$
  $t \leftarrow 0$
  $n \leftarrow 0$
  $\mathbf{sk} \leftarrow ()$
  $\mathsf{H}_{\mathsf{LIST}} \leftarrow \emptyset$
  $\mathsf{FL} \leftarrow \emptyset$
  $(j, K) \leftarrow \mathcal{A}^{\mathsf{New},\mathsf{H},\mathsf{Enc},\mathsf{Dec}}(\mathbf{pk})$
  if $K = K_j$
    return 1
  return 0
```

```
New():
  $t \leftarrow t + 1$
  $(\mathrm{pk}_t, \mathrm{sk}_t) \leftarrow \mathsf{Kg}(\mathrm{pars})$
  Append $\mathrm{sk}_t$ to $\mathbf{sk}$
  return $\mathrm{pk}_t$

Enc($i$):
  $n \leftarrow n + 1$
  $(C, K) \leftarrow \mathsf{KEM.encap}_{\mathrm{pk}_i}()$
  $\mathsf{FL} \leftarrow \mathsf{FL} \cup \{(C, i)\}$
  $K_n \leftarrow K$
  return $C$
```

```
H($K$):
  if $(K, h_K) \in \mathsf{H}_{\mathsf{LIST}}$
    return $h_K$
  else
    $h_K \xleftarrow{\$} \{0,1\}^\lambda$
    $\mathsf{H}_{\mathsf{LIST}} \leftarrow \mathsf{H}_{\mathsf{LIST}} \cup \{(K, h_K)\}$
    return $h_K$

Dec($C, i$):
  if $\{(C, i)\} \in \mathsf{FL}$ then
    return $\lightning$
  $K \leftarrow \mathsf{KEM.decap}_{\mathrm{sk}_i}(C)$
  return $K$
```

Figure 12: *The μOW-CCA security experiment (in the random oracle model). $\mathcal{A}$ can obtain as many key encapsulations as desired (specifying an index i for the public key it wishes to receive an encryption under) and needs to predict correctly the key of one of the encapsulations. $\mathcal{A}$ can decapsulate values of C, under specified secret key sk_i, as long as such a pair was not created by* Enc. *The normal* OW-CCA *game limits the number of* Enc *queries to one.*

Definition A.2. Let $\mathsf{DEM} = (\mathsf{DEM.Pg}, \mathsf{DEM.Kg}, \mathsf{DEM.Enc}, \mathsf{DEM.Dec})$ be a data encapsulation mechanism. Then the IND-CCA advantage of an adversary $\mathcal{A}$ against DEM is defined by

$$\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{DEM},\,\mathcal{A}}(\lambda) \overset{def}{=} \mathbf{Pr}\left[\mathbf{Exp}^{\mathsf{IND\text{-}CCA}}_{\mathsf{DEM},\,\mathcal{A}}(\lambda) = 1\right]$$

where the experiment $\mathbf{Exp}^{\mathsf{IND\text{-}CCA}}_{\mathsf{DEM},\,\mathcal{A}}(\lambda)$ is given in Fig. 13.

```
$\mathbf{Exp}^{\text{IND-CCA-}b}_{\mathsf{DEM},\ \mathcal{A}}(\lambda)$:
  pars $\leftarrow$ DEM.Pg$(1^\lambda)$
  FL $\leftarrow \emptyset$
  $K \leftarrow$ DEM.Kg(pars)
  $b' \leftarrow \mathcal{A}^{\mathsf{LR}_b,\mathsf{Dec}}$(pars)
  return $b'$

$\mathsf{LR}_b(m_0, m_1)$:
  if $|m_0| \neq |m_1|$ then
    return $\lightning$
  $\psi \leftarrow$ DEM.Enc$_K(m_b)$
  FL $\leftarrow$ FL $\cup \{\psi\}$
  return $\psi$

$\mathsf{Dec}(\psi)$:
  if $\psi \in$ FL then
    return $\lightning$
  $m \leftarrow$ DEM.Dec$_K(\psi)$
  return $m$
```

Figure 13: *The* IND-CCA *security experiments for* DEM.

B IND-PKDM-CCA is Equivalent to IND-CCA

For our main result, Theorem 3.1, we introduced and used a restricted KDM security of a DEM, called IND-PKDM-CCA security (Fig. 3). This notion is in fact equivalent to the standard IND-CCA security of the DEM. That IND-PKDM-CCA security implies IND-CCA security follows from standard relations between different formulations of IND-CCA security, plus the fact that a non-key dependent message can be queried (in the KDM world) by using a constant function. We show the reverse direction, namely that IND-CCA security for DEMs implies the IND-PKDM-CCA notion, by a hybrid argument.

Theorem B.1. Let DEM be a data encapsulation mechanism. Then for adversary $\mathcal{A}_1$, there exists an algorithm $\mathcal{A}_2$ of comparable computational complexity such that

$$\mathbf{Adv}^{\text{IND-PKDM-CCA}}_{\mathsf{DEM},\ \mathcal{A}_1}(\lambda) \leq n \cdot \mathbf{Adv}^{\text{IND-CCA}}_{\mathsf{DEM},\ \mathcal{A}_2}(\lambda)\ .$$

Proof. We seek a contradiction, by assuming that DEM is IND-CCA secure but not IND-PKDM-CCA secure, so there exists an algorithm $\mathcal{A}_1$ that breaks IND-PKDM-CCA. If we have $n+1$ keys in the system, then we have $n+1$ hybrid experiments Hyb-t as described below (for $t \in \{0, ..., n\}$).

```
$\mathbf{Exp}^{\text{Hyb-}t}_{\mathsf{DEM},\ \mathcal{A}}(\lambda)$:
  pars $\leftarrow$ DEM.Pg$(1^\lambda)$
  $i \leftarrow 0$
  FL $\leftarrow \emptyset$
  $b' \leftarrow \mathcal{A}^{\mathsf{New},\mathsf{LR}_b,\mathsf{Dec}}$(pars)
  return $b'$

$\mathsf{LR}_b(j, \vartheta)$:
  if $j \notin [i]$ then
    return $\lightning$
  $m_1 \leftarrow \vartheta(K^{j-1})$
  $m_0 \leftarrow 0^{|m_1|}$
  if $j \leq t$ then
    $\psi \leftarrow$ DEM.Enc$_{K_j}(m_1)$
  else
    $\psi \leftarrow$ DEM.Enc$_{K_j}(m_0)$
  FL $\leftarrow$ FL $\cup \{(j, \psi)\}$
  return $\psi$

$\mathsf{New}()$:
  $i \leftarrow i + 1$
  $K_i \leftarrow$ DEM.Kg(pars)
  return $i$

$\mathsf{Dec}(j, \psi)$:
  if $(j, \psi) \in$ FL then
    return $\lightning$
  $m \leftarrow$ DEM.Dec$_{K_j}(\psi)$
  return $m$
```

Figure 14: *The hybrid games* Hyb-t *for a* DEM. *Again* $\varphi(K^{i-1})$ *indicates function can depend on all keys in range* $\{K_0, ..., K_{i-1}\}$.

In the $t = 0$ hybrid we have $j > 0$ $(\forall j)$ so this refers to IND-PKDM-CCA-0, and for $t = n$ we have $j \leq t$ $(\forall j)$ which always returns an encryption of m_1 corresponding to IND-PKDM-CCA-1, which gives rise to equation (14).

$$\begin{aligned}
\mathbf{Adv}^{\text{IND-PKDM-CCA}}_{\mathsf{DEM},\ \mathcal{A}_1}(\lambda) &\overset{def}{=} \left|\mathbf{Pr}\left[b \leftarrow \{0,1\}; \mathbf{Exp}^{\text{IND-PKDM-CCA-}b}_{\mathsf{DEM},\ \mathcal{A}_1}(\lambda) = b\right] - 1/2\right| && (13)\\
&= \left|\mathbf{Pr}\left[\mathbf{Exp}^{\text{IND-PKDM-CCA-1}}_{\mathsf{DEM},\ \mathcal{A}_1}(\lambda) = 1\right] - \mathbf{Pr}\left[\mathbf{Exp}^{\text{IND-PKDM-CCA-0}}_{\mathsf{DEM},\ \mathcal{A}_1}(\lambda) = 1\right]\right| \\
&= \left|\mathbf{Pr}\left[\mathbf{Exp}^{\text{Hyb-}n}_{\mathsf{DEM},\ \mathcal{A}}(\lambda) = 1\right] - \mathbf{Pr}\left[\mathbf{Exp}^{\text{Hyb-0}}_{\mathsf{DEM},\ \mathcal{A}}(\lambda) = 1\right]\right| && (14)\\
&\leq n \cdot \mathbf{Adv}^{\text{IND-CCA}}_{\mathsf{DEM},\ \mathcal{A}_2}(\lambda) && (15)
\end{aligned}$$

Since we made the assumption that $\mathcal{A}_1$ breaks IND-PKDM-CCA, $\mathcal{A}_1$ can distinguish at least one gap between two hybrids in the sum. If we assume that $\mathcal{A}_1$ can distinguish between Hyb-t^* and

Hyb-(t^*-1) for some t^*, then we need to build a reduction $\mathcal{A}_2$ that can simulate all queries, and thus can compute m_1 (and subsequently m_0) by itself, and feed into the IND-CCA oracle and win the game. That is to say, $\mathcal{A}_1$'s advantage in distinguishing between Hyb-t^* and Hyb-(t^*-1) is greater than $1/n \cdot \mathbf{Adv}^{\mathsf{IND\text{-}PKDM\text{-}CCA\text{-}}b}_{\mathsf{DEM},\,\mathcal{A}_1}(\lambda)$. Utilising this fact, we create an algorithm $\mathcal{A}_2$ attacking the IND-CCA property of DEM, as detailed below, to prove equation (15).

```
$\mathcal{A}_2$ playing $\mathbf{Exp}^{\mathsf{IND\text{-}CCA\text{-}}b}_{\mathsf{DEM},\,\mathcal{A}_2}(\lambda)$:
  $\mathrm{pars} \leftarrow \mathsf{DEM.Pg}(1^\lambda)$
  for $j \in [n] \setminus \{t^*\}$ do $K_j \leftarrow \mathsf{DEM.Kg}(\mathrm{pars})$
  $b' \leftarrow \mathcal{A}_1^{\mathsf{LR}_b,\mathsf{Dec}}(\mathrm{pars})$
  return $b'$
```

```
$\mathsf{LR}_b(j, \vartheta)$:
  if $j < t^*$ then
      $m_1 \leftarrow \vartheta(K^{j-1})$
      $\psi \leftarrow \mathsf{DEM.Enc}_{K_j}(m_1)$
  if $j > t^*$ then
      $m_0 \leftarrow 0^{|m_1|}$
      $\psi \leftarrow \mathsf{DEM.Enc}_{K_j}(m_0)$
  if $j = t^*$ then
      $m_1 \leftarrow \vartheta(K^{j-1})$
      $m_0 \leftarrow 0^{|m_1|}$
      call $\psi \leftarrow \mathsf{LR}(m_0, m_1)$
  return $\psi$
```

```
$\mathsf{Dec}(\psi)$:
  call $m \leftarrow \mathsf{Dec}\ \psi$
  return $m$
```

Figure 15: *Description of reduction $\mathcal{A}_2$ used to prove (15). When $\mathcal{A}_2$ runs $\mathcal{A}_1$, it needs to create an environment $\mathbf{Exp}^{\mathsf{IND\text{-}PKDM\text{-}CCA\text{-}}b}_{\mathsf{DEM},\,\mathcal{A}_1}$. The "**call** ψ" line details that $\mathcal{A}_1$ calls the* IND-CCA *oracle* LR, *receiving an encryption of m_b under the correct* IND-CCA *challenge key K. The line "**call** m" indicates that $\mathcal{A}_1$ calls the decryption oracle of the* IND-CCA *game on ψ.*

For $j < t^*$ it is possible for the reduction simlulate m_1 correctly using K^{j-1}, and for $j > t^*$ we use the length regularity condition on ϑ to create the m_0 value to feed into the hybrids. Since we know $\mathcal{A}_1$ can beat IND-PKDM-CCA for $j = t^*$, we make the input values to the IND-CCA game's LR oracle be the same as the m_1 and m_0 values used in the IND-PKDM-CCA game, ensuring that the reduction captures this correctly.

□

C IND-KDM Security of TYPE-2 Hybrid Encryption

Theorem C.1. Let Hyb be a hybrid PKE scheme as defined above that comprises an OW-CCA TYPE-2 KEM and an IND-CCA DEM. For any adversary $\mathcal{A}$ that asks at most q oracle queries (encryption queries + direct RO queries + indirect RO queries), and for all length-regular $\varphi \in \Phi$, there exists algorithms $\mathcal{B}$ and $\mathcal{C}$ such that

$$\mathbf{Adv}^{\mathsf{IND\text{-}KDM\text{-}CCA}[\Phi]}_{\mathsf{Hyb},\,\mathcal{A}}(\lambda) \leq 2q \cdot \mathbf{Adv}^{\mathsf{OW\text{-}CCA}}_{\mathsf{KEM},\,\mathcal{B}}(\lambda) \;+\; 2\mathbf{Adv}^{\mathsf{IND\text{-}PKDM\text{-}CCA}}_{\mathsf{DEM},\,\mathcal{C}}(\lambda) + 4\mathbf{Adv}^{\mathsf{PRF}}_{F,\,\mathcal{A}}(\lambda).$$

Proof. The proof is identical to the proof of Theorem 3.1 up until equation (11), and instead we have:

$$\Pr[\mathcal{A} \text{ sets bad in } \mathsf{G}_3] \leq q \cdot \mathbf{Adv}^{\mathsf{OW\text{-}CCA}}_{\mathsf{KEM},\,\mathcal{B}}(\lambda) \tag{16}$$

In TYPE-2 KEMs an adversary playing the OW-CCA game cannot check if each query is equal to the challenge. As a result, in the simulation of LR_0 the probability that we can win the KEM game when $\mathcal{A}$ sets bad in G_2 is $\frac{1}{q}$, where q is the total number of queries that $\mathcal{A}$ makes. This factor carries through in the term collection at the end of the proof. □