Towards Key-Dependent Message Security in the Standard Model*

Dennis Hofheinz[1] and Dominique Unruh[2]

[1] CWI, Cryptology and Information Security Group, Amsterdam, The Netherlands, Dennis.Hofheinz@cwi.nl
[2] Saarland University, Information Security and Cryptology Group, Saarbrücken, Germany, unruh@cs.uni-sb.de

January 13, 2008

Abstract. Standard security notions for encryption schemes do not guarantee any security if the encrypted messages depend on the secret key. Yet it is exactly the stronger notion of security in the presence of *key-dependent* messages (KDM security) that is required in a number of applications: most prominently, KDM security plays an important role in analyzing cryptographic multi-party protocols in a formal calculus. But although often assumed, the mere existence of KDM secure schemes is an open problem. The only previously known construction was proven secure in the random oracle model.
We present symmetric encryption schemes that are KDM secure in the standard model (i.e., without random oracles). The price we pay is that we achieve only a relaxed (but still useful) notion of key-dependent message security. Our work answers (at least partially) an open problem posed by Black, Rogaway, and Shrimpton. More concretely, our contributions are as follows:

1. We present a (stateless) symmetric encryption scheme that is information-theoretically secure in face of a *bounded* number and length of encryptions for which the messages depend in an arbitrary way on the secret key.
2. We present a stateful symmetric encryption scheme that is computationally secure in face of an arbitrary number of encryptions for which the messages depend only on the respective *current* secret state/key of the scheme. The underlying computational assumption is minimal: we assume the existence of one-way functions.
3. We give evidence that the only previously known KDM secure encryption scheme cannot be proven secure in the standard model (i.e., without random oracles).

Keywords: Key-dependent message security, security proofs, symmetric encryption schemes.

1 Introduction

Proofs of security are a good and sound way to establish confidence in an encryption system. However, "proof" is a bit misleading here: usually, a security proof is not an absolute statement, but merely shows that *under certain assumptions*, the scheme is resistant against a *certain class of attacks*. Nothing is guaranteed if the assumptions are invalidated or attacks outside the considered class take place. Therefore, it is crucial that

- the underlying assumptions are plausible, and
- the considered class of attacks is as general as possible.

* To be published at Eurocrypt 2008.

Additionally, encryption schemes are most often used only as a building block in a larger protocol context, and thus

– the considered class of attacks should allow for meaningful and general analysis of the encryption scheme in a larger protocol context.

Indistinguishability of ciphertexts. The most established class of attacks consists of attacks targeted against the *indistinguishability of ciphertexts* (IND-CPA [16], resp. IND-CCA [21] attacks). Here, adversary A's goal is to win the following game: first, A chooses two messages m_0, m_1, then gets the encryption c_b of m_b (for a random $b \in \{0,1\}$), and finally outputs a guess b' for b. Now A wins if $b = b'$, i.e., if it guessed correctly which message was encrypted. The scheme is secure if no adversary wins (significantly) more often than in half of the cases. Intuitively, security in this sense implies that "one ciphertext looks like any other."

The IND-CPA and IND-CCA notions have been tremendously successful and even proved equivalent to a number of alternative and arguably not less appealing notions (cf. [5,6,10,19]). At the same time, IND-CPA and IND-CCA security can be achieved under various plausible number-theoretic assumptions [16,13,11].

Key-dependent message security. However, there is one security property that is useful and important in many applications, yet is *not* covered by IND-CPA or IND-CCA security: security in presence of *key-dependent* messages. More concretely, imagine a scenario in which the adversary can request encryptions of *arbitrary (but efficiently evaluatable) functions of the secret decryption key.* In other words, the adversary chooses a function g and gets the encryption of $g(K)$ under secret key K. Note that this is something the adversary may not be able to generate on its own, not even in the public-key setting. The adversary's goal is now to distinguish such a key-dependent encryption from an encryption of a random message. Security of an encryption is a *useful* notion to consider since

– in relevant practical settings, this notion is necessary: consider, e.g., encrypting your hard drive (which may contain the secret key, e.g., on the swap partition, or in a file that contains your secret keyring),
– certain protocols use key-dependent message security explicitly as a technical tool [8],

and, possibly most importantly from a theoretical perspective,

– key-dependent message security is a key ingredient for showing that security results that are proven in a formal calculus are also computationally sound.

This latter reason may come a bit surprising, hence we explain it in more detail.

Formal security proofs. The idea to automate security proofs can be traced back to the seminal work of Dolev and Yao [14], who described a formal calculus to analyze security protocols. To make the calculus accessible to automatic provers, however, base primitives like encryption (or, later, signatures) had to be over-idealized, disconnecting them from their concrete computational implementations. What was missing for almost 20 years was a soundness result, i.e., a result that essentially states "whatever can be proven in the abstract calculus holds as well in the cryptographic world, where the ideal encryption operator is implemented with an encryption scheme."

But finally, the soundness result by Abadi and Rogaway [1] connected the formal, machine-accessible world with the cryptographic world. However, with standard encryption schemes, only a certain subset of possible protocols could be considered, namely those that only contain expressions which fulfil a certain "acyclicity" condition.[3] To achieve full generality, a stronger requirement (security in the presence of key-dependent messages) on the encryption scheme was needed. This is not a peculiarity of the approach of Abadi and Rogaway; similar problems occur in related approaches, e.g. [20,2,4]. In particular, Adão et al. [2] show that in a certain sense, key-dependent message security is a necessity for formal soundness.

Related work. Around the time when the need for key-dependent security had been realized, formal characterizations of the security notion were given in [8,7]. Moreover, [7] showed a simple symmetric encryption scheme to be secure with respect to their notion. However, their scheme was proven in the random oracle model, and the proof made heavy use of the "ideal" nature of the random oracle (more details on this in Section 3). Black et al. posed the question of achieving key-dependent security in the *standard* model.

Backes et al. [3] consider several strengthenings of the definition from [7]. They prove structural results among the notions (including a way to "patch" a scheme that is secure in the sense of [7] to match the notions from [3]). However, Backes et al. do not give an actual construction of a secure scheme.

Our work. Our goal is to achieve key-dependent message security, as defined by Black et al., in the standard model. We present several results:

- a (stateless) symmetric encryption scheme that is information-theoretically secure in face of a *bounded* number and length of encryptions for which the messages depend in an arbitrary way on the secret key.
- a stateful symmetric encryption scheme that is computationally secure in face of an arbitrary number of encryptions for which the messages depend only on the respective *current* secret state/key of the scheme. The underlying computational assumption is minimal: we assume the existence of one-way functions.

We also stress the strictness of key-dependent message security:

- We give evidence that the only previously known KDM secure encryption scheme cannot be proven secure in the standard model (i.e., without random oracles).[4]

Note. Recently, we learned about the (concurrent and independent) work [17] of Halevi and Krawczyk. They are interested more generally in keyed primitives (such as pseudorandom functions, PRFs) which are secure in face of key-dependent inputs. They also show that an encryption scheme constructed from such a PRF inherits the underlying PRF's resilience against key-dependent inputs/messages. In particular, Halevi and Krawczyk construct a PRF (and a corresponding encryption scheme) that is secure in face of inputs which depend in an arbitrary, but *known-a-priori* way on the key. (That is, for each way in which the query may depend on the key, they give a PRF which is secure in face of such inputs.)

In contrast to that, we are interested in constructing encryption schemes that are secure in face of (encryptions of) messages that depend in an *arbitrary*, adaptively determined way on the key. Unfortunately, neither our schemes nor the schemes of [17] can handle the important case of non-trivial *key cycles*, that is, cyclic chains of encryptions of key K_i under key $K_{i+1 \bmod n}$.

[3] They also did only prove security against passive adversaries. However, active security was achieved by subsequently by [20,2,4].

[4] A similar, but technically different result is also contained in the independent work [17].

2 Preliminaries

Basic notation. Throughout the paper, $k \in \mathbb{N}$ denotes the *security parameter* of a given construction. Intuitively, a larger security parameter should provide more security, but a scheme's efficiency is also allowed to degrade with growing k. A *negligible* function vanishes faster than any given polynomial. The *statistical distance* between two random variables X and Y is denoted by $\delta(X\ ;\ Y)$. The *Rényi entropy* $H_2(X)$ of a random variable X is defined as $H_2(X) := -\sum_x \log_2 \mathsf{Pr}\left[X = x\right]^2$. Two families (X_k) and (Y_k) of random variables are *computationally indistinguishable* (written $X \approx Y$) if for every *PPT* (probabilistic polynomial-time) algorithm A, the function $|\mathsf{Pr}\left[A(X_k) = 1\right] - \mathsf{Pr}\left[A(Y_k) = 1\right]|$ is negligible in k. A family $\mathcal{UHF}$ of *universal hash functions* is a family of functions $h : \{0,1\}^n \to \{0,1\}^m$ with the property that for $x, x' \in \{0,1\}^n$ with $x \neq x'$, all $y, y' \in \{0,1\}^m$, and uniformly chosen $h \in \mathcal{UHF}$, we have that $\Pr[h(x) = y, h(x') = y'] = 2^{-2m}$.

We will further need a strengthened version of the leftover hash lemma that takes into account additional information S about the randomness K and some additional information Q unrelated to K.

Lemma 1 (Leftover Hash Lemma, extended). *Let K, Q, and S be random variables over bitstrings of fixed length. Let h be uniformly distributed over a family $\mathcal{UHF}$ of universal hash functions. Let U be uniformly distributed over bitstrings of length $|h(K)|$. Assume the following independences:*
- *U and (h, S, Q) are independent.*
- *K and Q are independent.*
- *h and (K, S, Q) are independent.*

Then the following bound holds:

$$\delta(h, h(K), S, Q\ ;\ h, U, S, Q) \ \leq\ 2^{|S|+|h(K)|/2 - H_2(K)/2 - 1}.$$

In a typical application of this lemma, h, K, and Q would be mutually independent, and S would be a function of (h, K, Q) (say, a side channel). Furthermore, U would be some completely independent random variable, representing the ideal randomness. This would then imply all the independence conditions in the lemma.

Proof. In the following, s, q, k range over all values taken by S, Q, K, respectively. By applying the definition of the statistical distance, we have

$$\begin{aligned} \varepsilon &:= \delta(h, h(K), S, Q\ ;\ h, U, S, Q) \\ &= \sum_{s,q} \Pr[S = s, Q = q]\, \delta(h, h(K)|S = s, Q = q\ ;\ h, U|S = s, Q = q). \end{aligned} \tag{1}$$

Here $X|(S = s)$ stands for the distribution of X under the condition $S = s$. Since h and (K, S, Q) are independent, $h|(S = s, Q = q)$ is a universal hash-function. And since U is independent of (S, Q, h), we have that U is uniformly distributed and independent of h given $S = s, Q = q$. Further, since by assumption h is independent of (K, S, Q), we have that h and K are independent given $S = s, Q = q$. Thus the leftover hash lemma in its basic form [18] applies, and we get

$$\delta(h, h(K)|S = s, Q = q\ ;\ h, U|S = s, Q = q) \ \leq\ 2^{|h(K)|/2 - H_2(K|(S=s,Q=q))/2 - 1}.$$

Combining this with (1) we get

$$
\begin{aligned}
\varepsilon &\leq \sum_{s,q} \Pr[S=s, Q=q] \cdot 2^{|h(K)|/2 - H_2(K|(S=s,Q=q))/2-1} \\
&= \sum_{s,q} \Pr[S=s, Q=q] \cdot \frac{1}{2}\sqrt{2^{|h(K)|} \cdot \sum_k \Pr[K=k|S=s,Q=q]^2} \\
&\leq \sum_{s,q} \Pr[Q=q] \cdot \frac{1}{2}\sqrt{2^{|h(K)|} \cdot \sum_k \Pr[S=s|Q=q]^2 \cdot \Pr[K=k|S=s,Q=q]^2} \\
&= \sum_{s,q} \Pr[Q=q] \cdot \frac{1}{2}\sqrt{2^{|h(K)|} \cdot \sum_k \Pr[K=k, S=s|Q=q]^2} \\
&\leq \sum_{s,q} \Pr[Q=q] \cdot \frac{1}{2}\sqrt{2^{|h(K)|} \cdot \sum_k \Pr[K=k|Q=q]^2} \\
&\overset{(*)}{=} \sum_{s,q} \Pr[Q=q] \cdot \frac{1}{2}\sqrt{2^{|h(K)|} \cdot \sum_k \Pr[K=k]^2} \\
&= \sum_{s,q} \Pr[Q=q] \cdot \frac{1}{2}\sqrt{2^{|h(K)|} \cdot 2^{-H_2(K)}} \\
&= \sum_{s,q} \Pr[Q=q] \cdot 2^{|H(k)|/2 - H_2(K) - 1} \\
&= \sum_{s} 2^{|H(k)|/2 - H_2(K) - 1} = 2^{|S| + |H(k)|/2 - H_2(K) - 1}.
\end{aligned}
$$

Here $(*)$ uses that Q and K are independent. □

Key-dependent message security. For formalizing key-dependent message security, we use a variation on the definition of Black et al. [7]:

Definition 2 (KDM security, standard model, symmetric setting). *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme, let $\mathbf{K} := (K_1, \ldots, K_n)$ be secret keys (where n is polynomial in the security parameter), and let A be an adversary. Let*

- $\mathsf{Real}_{\mathbf{K}}$ *be the oracle that on input g, μ returns $C \leftarrow \mathcal{E}(1^k, K_\mu, g(\mathbf{K}))$, and*
- $\mathsf{Fake}_{\mathbf{K}}$ *be the oracle that on input g, μ returns $C \leftarrow \mathcal{E}(1^k, K_\mu, U)$ for an independently uniformly selected fresh $U \in \{0,1\}^{|g(\mathbf{K})|}$.*

In both cases, g is encoded as a circuit.[5] *The* KDM advantage *of A is*

$$\mathsf{Adv}_{\Pi}^{KDM}(A) := \left| \mathsf{Pr}\left[\mathbf{K} \xleftarrow{\$} \mathcal{K} : A^{\mathsf{Real}_{\mathbf{K}}(\cdot)} = 1\right] - \mathsf{Pr}\left[\mathbf{K} \xleftarrow{\$} \mathcal{K} : A^{\mathsf{Fake}_{\mathbf{K}}(\cdot)} = 1\right] \right|$$

Here $\mathbf{K} \xleftarrow{\$} \mathcal{K}$ means that each key K_i is chosen independently using $\mathcal{K}$.

We say that Π is KDM secure *iff for every PPT adversary A and every polynomial n, the advantage function $\mathsf{Adv}_{\Pi}^{KDM}(A)$ is negligible in the security parameter. We require that A only queries its oracle with fixed-length functions g, i.e., $|g(K)|$ is the same for all values of K.*

[5] This has the side-effect that for a polynomial-time adversary A, the function g is also polynomial-time computable.

The relation to real-or-random security. Definition 2 bears a great resemblance to the real-or-random (ROR-CPA) definition for encryption schemes from [5]. The main difference is that Definition 2 equips the adversary with an oracle that delivers encryptions of *key-dependent* messages (i.e., evaluations) $g(K)$. The way in which these messages depend on the keys is completely up to the adversary; the only constraint is that g must be efficiently evaluatable and have a fixed output length.

On achieving KDM security and active KDM security. Using the equivalence of ROR-CPA and IND-CPA security from [5], it is easy to see that Definition 2 is *strictly* stronger than IND-CPA security. A natural adaption of Definition 2 to active attacks—such a notion is called AKDM security in [3]—consists in equipping the adversary with a decryption oracle that is restricted in the usual sense to prevent trivial attacks. And similarly to the passive case, it is easy to see that AKDM security is *strictly* stronger than IND-CCA security. On the other hand, once a scheme is KDM secure, it can be easily and without (much) loss of efficiency upgraded to AKDM security, as formalized and proved in [3]. Hence, the main difficulty lies in finding a scheme that is KDM secure in the first place. In the following, this will be our focus.

3 The scheme of Black et al.

Definition 2 is very hard to achieve. In fact, the only construction that is known, due to Black et al. [7], to achieve Definition 2 is in the random oracle model. It will be very useful to take a closer look at their scheme. We will argue that in a very concrete sense, nothing less than a random oracle will do for their scheme. Hence, their construction merely shows how powerful random oracles are, but does not give a hint on how to achieve KDM security in the standard model. This constitutes one motivation for our upcoming weakening of KDM security.

Scheme 3 (The scheme ver). Define the symmetric encryption scheme **ver** $= (\mathcal{K}, \mathcal{E}, \mathcal{D})$ with security parameter $k \in \mathbb{N}$, message space $\{0,1\}^k$ and key space $\{0,1\}^k$ through
- $\mathcal{K}(1^k)$ outputs a uniform random key $K \in \{0,1\}^k$.
- $\mathcal{E}(1^k, K, M)$ samples $R \overset{\$}{\leftarrow} \{0,1\}^k$ and outputs the ciphertext $(R, H(K||R) \oplus M)$.
- $\mathcal{D}(1^k, K, (R, D))$ outputs the message $H(K||R) \oplus D$.

The security of ver with a random oracle. Black et al. prove

Theorem 4 (Security of ver [7]). *If H is a random oracle, then* **ver** *is KDM secure.*

The main idea of the proof is to consider an event **bad**, where **bad** occurs iff
1. the adversary queries H at any point $K||R$ that was *previously* used for encryption, or
2. one of the functions g submitted to the encryption oracle queries H at the *currently used* point $K||R$.

If **bad** does not occur, the adversary's view is *identical* in the Real and Fake experiments, thanks to the fact that different random oracle queries $H(X), H(Y)$ $(X \neq Y)$ are statistically independent: each message is padded with *completely fresh* and message-independent randomness. Hence, by showing (with an inductive argument) that **bad** occurs only with small probability, [7] show the scheme **ver** KDM secure.

The insecurity of ver without a random oracle. Put informally, the proof of **ver** utilizes one essential property of the random oracle H: knowledge about arbitrary many values $H(Y_i)$ (with $Y_i \neq X$) does not yield *any* information about $H(X)$. This use of a random oracle as a provider of statistical independence is what makes the proof fail completely with any concrete hash function used in place of the random oracle. There is no hope for the proof strategy to succeed without random oracles. A little more formally, we can show that in the random oracle model, there exists a specific hash function H that has a number of generally very useful properties: H is collision-resistant, one-way, can be interpreted as a pseudorandom function (in a way compatible with **ver**), and H makes **ver** IND-CPA. *But* H makes **ver** completely insecure in the presence of key-dependent messages. Hence, there can be no fully black-box KDM security proof for **ver** that relies on these properties of H alone.

Theorem 5 (Insecurity of ver). *Relative to a random oracle $\mathcal{O}$, there exists a function H such that*

1. *H is collision-resistant,*
2. *for any function $p(k) \in k^{\Theta(1)}$, H is one-way w.r.t. the uniform input distribution on $\{0,1\}^{p(k)}$,*
3. *the function $F_K(R) := H(K||R)$ is a pseudorandom function with seed K,*
4. *the scheme* **ver***, instantiated with H, is IND-CPA secure, but*
5. *the scheme* **ver***, instantiated with H, is not KDM secure.*

Proof (sketch). Assume for simplicity that the security parameter k is even. Say that the random oracle $\mathcal{O}$ maps arbitrary bitstrings to k-bit strings. Then denote by $\mathcal{O}_\ell(x)$ the first $k/2$ bits of $\mathcal{O}(x)$. Now consider the function $H : \{0,1\}^* \to \{0,1\}^k$ with

$$H(x) := \begin{cases} \mathcal{O}(x) & \text{for } |x| \neq 2k, \\ \mathcal{O}(x_\ell) \oplus (\mathcal{O}_\ell(x)||\mathcal{O}_\ell(\mathcal{O}_\ell(x))) & \text{for } x = x_\ell||x_r \text{ and } |x_\ell| = |x_r| = k. \end{cases}$$

We show the claimed properties for H:

1. H is collision-resistant. It is clear that collisions $H(x) = H(y)$ (with $x \neq y$) cannot be found efficiently if $x \neq 2k$ or $y \neq 2k$. So assume $x = x_\ell||x_r$ and $y = y_\ell||y_r$ for $|x_\ell| = |x_r| = |y_\ell| = |y_r| = k$. Collisions of this form imply $\mathcal{O}_\ell(x_\ell) \oplus \mathcal{O}_\ell(x) = \mathcal{O}_\ell(y_\ell) \oplus \mathcal{O}_\ell(y)$ and thus

$$\mathcal{O}_\ell(x_\ell) \oplus \mathcal{O}_\ell(y_\ell) = \mathcal{O}_\ell(x) \oplus \mathcal{O}_\ell(y). \tag{2}$$

If $x_\ell = y_\ell$, then this constitutes a collision in $\mathcal{O}_\ell$, so we may assume $x_\ell \neq y_\ell$. But the distributions of $\mathcal{O}_\ell$ on k-bit strings and on $2k$-bit strings are independent and both uniform. Hence, finding x and y to satisfy (2) requires a superpolynomial number of queries to $\mathcal{O}_\ell$ (resp. $\mathcal{O}$) with overwhelming probability.

2. H is one-way w.r.t. the uniform distribution on $\{0,1\}^k$. For $p(k) = 2k$, this follows from collision-resistance and the fact that H is compressing: Since the preimages of H are not unique, if we are able to find a preimage x' of $H(x)$ for random $x \in \{0,1\}^{2k}$, with noticeable probability we will have $x \neq x'$. This allows to find collisions efficiently. For details see [12]. For $p(k) \neq 2k$, this follows by definition of H and the fact that the random oracle is one-way.

3. $F_K(R) := H(K||R)$ **is a pseudorandom function.** Consider an adversary A that has oracle access to $\mathcal{O}$ and to F_K for uniformly chosen K. We denote A's i-th query to F_K by R_i. Without loss of generality, assume that A never asks for the same F_K evaluation twice, so the R_i are pairwise distinct. Furthermore, let $X_i := K||R_i$, and $Y_i := \mathcal{O}_\ell(K||R_i)$. We claim that A doesn't query $\mathcal{O}$ with K or *any* of the values X_i, Y_i, except with negligible probability.

We prove our claim inductively as follows. Let E_i denote the event that A queries $\mathcal{O}$ with a value that starts with K *prior* to the i-th F_K query. Clearly, E_1 happens with exponentially small probability. So fix an $i \geq 1$. To complete our proof, it is sufficient to show that under condition $\neg E_i$, the probability for E_{i+1} to happen is bounded by a negligible function that does not depend on i.

Assume that $\neg E_i$ holds. That means that, given A's view up to and including the $(i-1)$-th F_K query, the key K is uniformly distributed among all k-bit values (or k-bit prefixes of $2k$-bit values) not yet queried by A. By the polynomiality of A, this means that, from A's point of view, K is uniformly distributed on an exponentially-sized subset of $0, 1^k$. But this means that until the i-th F_K query, A has only an exponentially small chance to query one of K, X_j, Y_j $(j < i)$. Hence $E_{i+1} \mid \neg E_i$ happens only with exponentially small probability.

Summing up, A never queries $\mathcal{O}$ with K or any of the X_i, Y_i, except with negligible probability. Hence, F_K can be substituted with a truly random function without A noticing, and the claim follows.

4. ver with H **is IND-CPA.** Follows immediately from 3.

5. ver with H **is not KDM secure.** A successful KDM adversary A on **ver** is the following: A asks its encryption oracle for an encryption of $\mathcal{O}(K)$ (e.g., using g with $g(x) = \mathcal{O}(x)$ as input to the oracle). In the real KDM game, the ciphertext will be

$$(R, H(K||R) \oplus \mathcal{O}(K)) = (R, \mathcal{O}_\ell(K||R)||\mathcal{O}_\ell(\mathcal{O}_\ell(K||R))),$$

and hence of the form $(R, t||\mathcal{O}_\ell(t))$ for some t, which can be easily recognized by A. But in the fake KDM game, the ciphertext will have the form (R, U) for a uniformly and independently distributed U, which is generally not of the form $(R, t||\mathcal{O}_\ell(t))$. Hence, A can successfully distinguish real encryptions from fake ones. $\square$

Halevi and Krawczyk's example. Halevi and Krawczyk give a different example of the "non-implementability" of **ver** (see [17, Negative Example 4]). They argue that the random oracle H in **ver** cannot be implemented with a PRF that is constructed from an ideal cipher using the Davies-Meyer transform. Their example has the advantage of being less artificial, while being formulated in the ideal cipher model.

4 Information-theoretic KDM security

Since key-dependent message security is very hard to achieve, we start with two simple schemes that do not achieve full KDM security, but serve to explain some important concepts.

4.1 The general idea and a simple scheme (informal presentation)

First observe that the usual one-time pad $C = M \oplus K$ (where C is the ciphertext, M the message, and K the key) does *not* achieve KDM security. Encryption of $M = K$ results in an all-zero ciphertext that is clearly distinguishable from a random encryption. However, the slight tweak

$$C = (h, M \oplus h(K)) \quad (h \text{ independently drawn universal hash function})$$

does achieve a certain form of key-dependent message security: the pad $h(K)$ that is distilled from K is indistinguishable from uniform and independent randomness, even if h and some arbitrary (but bounded) information $M = M(K)$ about K is known. (When using suitable bitlengths $|K|$ and $|M|$, this can be shown using the leftover hash lemma [18].) So the encryption $M \oplus h(K)$ of one single message $M = M(K)$ looks always like uniform randomness. Hence the scheme is KDM secure in a setting where the encryption oracle is only used once (but on the other hand, information-theoretic security against unbounded adversaries is achieved).

4.2 A more formal generalization of the simple scheme

Of course, one would expect that by expanding the key, the scheme stays secure even after *multiple* (key-dependent) encryptions. This is true, but to show this, a hybrid argument and multiple applications of the leftover hash lemma are necessary. We formalize this statement now.

Scheme 6 (The scheme p-BKDM (for "p-bounded KDM")). Let $p \in \mathbb{Z}[k]$ be a positively-valued polynomial, let $\ell(k) := (2p(k)+3)k$, and let $\mathcal{UHF}$ be a family of universal hash functions that map $\ell(k)$-bit strings to k-bit strings. Define the symmetric encryption scheme p-**BKDM** $= (\mathcal{K}, \mathcal{E}, \mathcal{D})$ with security parameter $k \in \mathbb{N}$, message space $\{0,1\}^k$, and key space $\{0,1\}^{\ell(k)}$ through

- $\mathcal{K}(1^k)$ outputs a uniform random key $K \in \{0,1\}^{\ell(k)}$.
- $\mathcal{E}(1^k, K, M)$ samples $h \xleftarrow{\$} \mathcal{UHF}$ and outputs the ciphertext $C = (h, h(K) \oplus M)$.
- $\mathcal{D}(1^k, K, (h, D))$ outputs the message $h(K) \oplus D$.

Definition 7 (Bounded KDM security). *Let $p \in \mathbb{Z}[k]$ be a positively-valued polynomial. Then a symmetric encryption scheme Π is* p-bounded KDM secure *if it is KDM secure against PPT adversaries that query the encryption oracle at most $p(k)$ times. Further, Π is* information-theoretically p-bounded KDM secure *if it is KDM secure against arbitrary (i.e., computationally unbounded) adversaries that query the encryption oracle at most $p(k)$ times.*

Theorem 8 (Bounded KDM security of p-BKDM). *The scheme p-**BKDM** is information-theoretically p-bounded KDM secure.*

Proof. In the following, we abbreviate $x_i, .., x_j$ with $x_{i..j}$ for all variables x. Let n be the number of keys used. Let an adversary A be given that queries the encryption oracle at most $p(k)$ times. Without loss of generality we can assume the adversary to be deterministic (by fixing the random tape that distinguishes best) and that it performs exactly $p(k)$ queries. In the i-th encryption in the real experiment, let μ_i denote the index of the key that has been used, let h_i be the hash function chosen by the encryption function, let m_i be the message that is encrypted, and let c_i be the second component of the resulting ciphertext (i.e., (h_i, c_i) is the i-th ciphertext). Since the adversary is deterministic, m_i depends deterministically from the keys $K_{1,n}$ and the ciphertexts $c_{1..i-1}, h_{1..i-1}$, i.e., there are deterministic functions $\hat{f}_i$ with $m_i = \hat{f}_i(K_{1,n}, c_{1..i-1}, h_{1..i-1})$. Similarly, there are deterministic functions $\hat{\mu}_i$ such that $\mu_i = \hat{\mu}_i(c_{1..i-1}, h_{1..i-1})$.

Let U_i be independent uniformly distributed random variables on $\{0,1\}^k$ that are independent of all random variables defined above. Let

$$\varepsilon_i := \delta(h_{1..i}, c_{1..i} \; ; \; h_{1..i}, U_{1..i})$$

To show that the scheme is information-theoretically p-bounded KDM secure, i.e., that the adversary cannot distinguish the real and the fake experiment, it is sufficient to show that $\varepsilon_{p(k)}$ is negligible since the view of A can be deterministically computed from $h_{1..p(k)}, c_{1..p(k)}$.

Fix some $i \in \{1, \dots, p(k)\}$. Let $K := K_{\mu_i}$, $Q := h_{1..i-1}$, $S := (m_i, c_{1..i-1})$, $h := h_i$ and let U be uniformly distributed on $\{0,1\}^k$ and independent of (K, Q, S, h). The following conditions hold by construction:

- h is a universal hash function.
- U is uniformly distributed and independent of (h, S, Q).
- K and Q are independent.
- h is independent of (K, S, Q).

So the conditions for Lemma 1 are fulfilled and we have

$$\delta(h, h(K), S, Q\ ;\ h, U, S, Q) \leq 2^{|S|+|h(K)|/2-H_2(K)/2-1} = 2^{ik+k/2-\ell(k)/2-1} \leq 2^{-k}$$

and thus

$$\begin{aligned}&\delta(h_{1..i}, c_i, c_{1..i-1}\ ;\ h_{1..i}, U_i, c_{1..i-1})\\ &\qquad\qquad \leq \delta(h_{1..i}, h_i(K_{\mu_i}), m_i, c_{1..i-1}\ ;\ h_{1..i}, U, m_i, c_{1..i-1}) \leq 2^{-k}\end{aligned} \tag{3}$$

Since (h_i, U_i) is independent of $(h_{1..i-1}, c_{1..i-1}, U_{1..i-1})$ by construction, from (4.2) we have

$$\delta(h_{1..i}, U_i, c_{1..i-1}\ ;\ h_{1..i}, U_i, U_{1..i-1}) = \varepsilon_{i-1}$$

and hence using (3) and the triangle inequality for the statistical distance, we have

$$\varepsilon_i = \delta(h_{1..i}, c_i, c_{1..i-1}\ ;\ h_{1..i}, U_i, U_{1..i-1}) \leq 2^{-k} + \varepsilon_{i-1}.$$

Since $\varepsilon_0 = 0$, it follows that $\varepsilon_{p(k)} \leq p(k) \cdot 2^{-k}$ is negligible. □

4.3 Discussion

The usefulness of bounded KDM security. Our scheme p-**BKDM** can be used in any protocol where the total length of the encrypted messages does not depend on the length of the key. At a first glance, this restriction seems to defeat our purpose to be able to handle key cycles: it is not even possible to encrypt a key with itself. However, a closer inspection reveals that key dependent messages occur in two kinds of settings. In the first setting, a protocol might make explicit use of key cycles in its protocol specification, e.g., it might encrypt a key with itself (we might call this *intentional key cycles*). In this case, p-**BKDM** cannot be used. In the second setting, a protocol does not explicitly construct key cycles, but just does not exclude the possibility that—due, e.g., to some leakage of the key—some messages turn out to depend on the keys (we might call this *unintentional key cycles*). In this case, the protocol does not itself construct key cycles (so the restriction of p-**BKDM** that a message is shorter than the key does not pose a problem), but only requires that *if key cycles occur* the protocol is still secure. But this is exactly what is guaranteed by p-**BKDM**. So for the—possibly much larger—class of protocols with unintentional key cycles the p-**BKDM** scheme can be used.

Multiple sessions of p-BKDM. Theorem 8 guarantees that even in the case of multiple sessions, the scheme p-**BKDM** is secure assuming that at most $p(k)$ encryptions are performed *in all sessions together*. In some applications, especially if the number of sessions cannot be bounded in advance, one might need the stronger property that we may encrypt $p(k)$ messages *with each key*. Intuitively, we might argue that when we receive an encryption $(h, M \oplus h(K))$ of a message M, the entropy of the key K decreases by at most $|M \oplus h(K)|$ bits, but as long as enough entropy remains in K, we do not learn anything about M, and neither about the keys M depends on. This leads to the following conjecture:

Conjecture 9. The scheme p-**BKDM** is KDM-secure if the adversary performs at most $p(k)$ encryptions under each key K_i. This holds even if different keys have different associated polynomials p_i (i.e., key K_i has length $O(p_i(k)k)$ and we encrypt p_i times under K_i).

Unfortunately, we do not know how to formally prove Conjecture 9. Formalizing the above intuition is not straightforward, since it is not clear how to alone define what it means that the entropy of a given key decreases while the entropy of the others does not. We leave this conjecture as an open problem.

Why encrypt *only* key-dependent messages? Definitions 2 and 7 give the adversary (only) access to an encryption oracle which encrypts arbitrary functions of the key (in contrast to [17] which additionally provides an encryption oracle for normal messages). In Definition 2, no generality is lost, since an ordinary encryption oracle can be emulated by choosing this function as a constant function. Call such "ordinary" encryption queries *non-KDM queries*. Now it is conceivable that a scheme allows for an unbounded number of non-KDM queries, but only a limited number of actually key-dependent queries. The security of such schemes can be appropriately captured using, e.g., the security definition of [17], which incorporates *separate* encryption oracles for key-dependent and non-KDM queries. While our Definition 7 does not allow to model such schemes, it is easy to see that our scheme p-**BKDM** is not secure against an unbounded number of non-KDM encryptions (not even against computationally bounded adversaries).

5 Computational KDM security

5.1 Motivation

The dilemma with hybrid arguments. The discussion in Section 4.3 does not only apply to our scheme p-**BKDM**. There seems to be a general problem with proving KDM security with a hybrid argument. Starting with the real KDM game, substituting the first encryption with a fake one first is not an option: the later encryptions cannot be properly simulated. But to substitute the last real encryption first is not easy either: for this, there first of all has to be a *guarantee* that at that point, the last key has not already leaked completely to the adversary. In our case, with a bounded overall number of encryptions, we can give an information-theoretic bound on the amount of information that has been leaked before the last encryption. But if there is no such bound, information theory cannot be used to derive such a bound. Instead, a computational assumption must be used. Yet, there seems to be no straightforward way to derive a useful statement (e.g., about the computational key leakage) that reaches across a polynomial number of instances from a single computational assumption *without* using a hybrid argument. Of course, this excludes certain interactive assumptions, which essentially already assume security of the scheme in the first place. We do not believe that it is useful or interesting to investigate such constructions and assumptions.

In other words, we cannot use hybrid arguments since we do not know where to place the first hybrid step. This situation is similar (but not identical) to the case of selective decommitments [15] and adaptively secure encryption (e.g., [9]).

Hybrid (KEM/DEM) encryption schemes. Another common tool for constructing encryption schemes are hybrid encryption schemes (no relation to hybrid arguments). In a hybrid encryption scheme, a ciphertext consists of a KEM (key encapsulation mechanism) part and a DEM (data encapsulation mechanism) part. The KEM part of the ciphertext encapsulates a symmetric key K that is unrelated to the message M to be encrypted. The DEM part of the ciphertext is a (symmetric) encryption of M under K. The actual secret key *sk* of the hybrid scheme is the secret key that is needed to decrypt the KEM part. It is tempting to use a hybrid construction to get rid of the dependency of message and secret key. However, there still is a dependency between M and *sk*: the KEM ciphertext provides a relation between *sk* and K on the one hand, and the DEM ciphertext relates K and M on the other. Hybrid encryption techniques do not help to get rid of dependencies between message and secret key.

Similarly, hybrid encryption techniques cannot be used to increase the allowed message lengths of the scheme from the previous section. Concretely, it may be tempting to use the p-**BKDM** scheme as a KEM to encapsulate a short key K, and then to use that key K as secret key for a *computationally secure* DEM which encrypts long messages with short keys. Unfortunately, this breaks the security proof of p-**BKDM** (and also, depending on the used DEM, also the security itself). Namely, the proof of p-**BKDM** depends not on the size of the KEM key K, but on the amount of released information about the actual KEM secret key (which corresponds to the length of the message in the KDM setting). So hybrid encryption does not help here, either.

Stateful KDM security. To nonetheless get a scheme that is secure in face of arbitrarily many encryptions of key-dependent messages, we propose *stateful* encryption schemes. In a stateful encryption scheme, the secret key (i.e., the internal state) is updated on each encryption. (Decryption must then be synchronized with encryption: we assume that ciphertexts are decrypted in the order they got produced by encryption.) For such a stateful encryption scheme, there are essentially two interpretations of KDM security:

- the message may depend on the current secret key (i.e., state) only, or
- the message may depend on the current and all previously used secret keys (i.e., on the current and all previous states).

We call the first notion *weak stateful KDM security*, and the second *strong stateful KDM security*. Weak stateful KDM security can be thought of as KDM security in a setting in which erasures are trusted, and strong stateful KDM security mandates that erasures are *not* trusted (in the most adversarial sense).

Definition 10 (Weak and strong stateful KDM security). *A stateful symmetric encryption scheme Π is secure in the sense of* weak stateful KDM security *iff Π is fulfills Definition 2, where the encryption queries are interpreted as a function in the* current *state of the encryption algorithm. Further, Π is secure in the sense of* strong stateful KDM security *iff Π satisfies Definition 2, where the encryption queries are interpreted as a function in the* current and all previous states *of the encryption algorithm.*

Below we will give a scheme that circumvents the hybrid argument dilemma using precisely the fact that there is a changing state.

Relation to Black et al.'s notion of "stateful KDM security". Black et al. [7] already consider the potential KDM security of a stateful symmetric encryption scheme. They show that there *can be no stateful KDM secure scheme*. However, they showed this under the assumption that encryption is deterministic. In our definition, encryption is still probabilistic, even though stateful. We use the state update mechanism *in addition* to using randomness, not instead of it. Their argument does not apply to our definition of stateful KDM security, neither to our weak nor to our strong variant.

Weak vs. strong stateful KDM security. For some applications, strong stateful KDM security is necessary: encrypting your hard drive (that may contain the secret key) cannot be done in a provably secure way with weak stateful KDM security. (Once the secret key gets to be processed by the scheme, the state may have already been updated, so that the message now depends on a *previous* state.) Also, the notion of *key cycles* (i.e., key K_i is encrypted under $K_{i+1 \bmod n}$) does not make sense with weak stateful KDM secure schemes. In these cases, the use of a *strong* stateful KDM scheme is fine. However, it seems technically much more difficult to construct a strong stateful KDM secure scheme.

5.2 A secure scheme

We do not know how to fulfill strong stateful KDM security. (The issues that arise are similar as in the stateless case.) However, we *can* present a scheme that is secure in the sense of weak stateful KDM security.

Idea of the construction. Our scheme is a computational variant of p-**BKDM** (although its analysis will turn out to be very different). The main problem of p-**BKDM** is that the secret key runs out of entropy after too many KDM encryptions. Only as long as there is enough entropy left in K, a suitably independent random pad can be distilled for encryption. However, in a computational setting, randomness can be expanded with a pseudorandom generator, and some distilled, high-quality randomness can be used to generate more (pseudo-)randomness as a new key. More concretely, consider the following scheme:

Scheme 11 (The scheme sKDM (for "stateful KDM")). Let $\mathcal{UHF}$ be a family of universal hash functions that map $5k$-bit strings to k-bit strings, and let G be a pseudorandom generator (against uniform adversaries) that maps a k-bit seed to a $6k$-bit string. Define the stateful symmetric encryption scheme $\mathbf{sKDM} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ with security parameter $k \in \mathbb{N}$, message space $\{0,1\}^k$, and key space $\{0,1\}^{5k}$ through

- $\mathcal{K}(1^k)$ outputs a uniform random initial key (i.e., state) $K_0 \in \{0,1\}^{5k}$.
- $\mathcal{E}(1^k, K_j, M_j)$ proceeds as follows:
  1. sample $h_j \stackrel{\$}{\leftarrow} \mathcal{UHF}$,
  2. set $S_j := h_j(K_j)$,
  3. set $(K_{j+1}, P_j) := G(S)$,
  4. output $C_j := (h_j, P_j \oplus M_j)$.

  Ciphertext is C_j, and new key (i.e., state) is K_{j+1}.
- $\mathcal{D}(1^k, K_j, (h_j, D_j))$ proceeds as follows:

1. set $S_j := h_j(K_j)$,
2. set $(K_{j+1}, P_j) := G(S)$,
3. output $M_j := P_j \oplus D_j$.

Plaintext is M_j, and new key (i.e., state) is K_{j+1}.

Theorem 12. *If G is a pseudorandom generator, then* **sKDM** *satisfies weak stateful KDM security.*

Proof. Fix an adversary A that attacks **sKDM** in the sense of weak stateful KDM security. Say that, without loss of generality, A makes precisely $p(k)$ encryption queries for a positively-valued polynomial $p \in \mathbb{Z}[k]$. Assume that A has an advantage that is not negligible.

Preparation for the hybrid argument. For $0 \leq j \leq p(k)$, define the hybrid game Game j as follows. Game j is the same as the weak stateful KDM game with adversary A, only that

- the first j encryption queries are answered as in the fake weak stateful KDM game (i.e., with encryptions of uniform and independent randomness), and
- the remaining queries are answered as in the real game (i.e., with encryptions of adversary-delivered functions evaluated at the *current* secret key).

Base step for the hybrid argument. We will reduce distinguishing between two adjacent games to some computational assumption. We will now first formulate this assumption. Let $K \in \{0,1\}^{5k}$ be uniformly distributed, and let $M \in \{0,1\}^k$ be arbitrary (in particular, M can be a function of K). Then by Lemma 1 it follows that $\delta(M, h, h(K)\ ;\ M, h, U_k) \leq 2^{-k}$ for independently sampled $h \xleftarrow{\$} \mathcal{UHF}$ and independent uniform $U_k \in \{0,1\}^k$. (Actually, in this case we could even use the original version of the Leftover Hash Lemma [18].) This implies

$$\delta(M, h, G(h(K))\ ;\ M, h, G(U_k)) \leq 2^{-k},$$

from which the computational indistinguishability chain

$$\underbrace{(M, h, G(h(K)))}_{=:D^R} \approx (M, h, G(U)) \approx \underbrace{(M, h, U_{6k})}_{=:D^F} \tag{4}$$

for independent uniform $U_{6k} \in \{0,1\}^{6k}$ follows by assumption on G. For our hybrid argument, it is important that (4) even holds when M is a function of K chosen by the distinguisher.

Hybrid argument. We will now construct from adversary A an adversary B that contradicts (4) by distinguishing D^R and D^F. This contradiction then concludes our proof. Let n denote the number of keys. Let μ_i denote the index of the key chosen by A for the i-th encryption. Let g_i denote the function chosen by A in the i-th encryption. Then, the adversary B chooses some $j \in \{1, \dots, p(k)\}$ uniformly at random and then performs the following simulation for A:

- The first $j-1$ encryptions requested by A are simulated as fake encryptions (i.e., with random messages). This is possible without using the keys since for a random message, $h_i(K_{\mu_i})$ is information-theoretically hidden in the ciphertext.
- For the j-th encryption, B chooses K_μ randomly for all $\mu \neq \mu_j$ and defines[6] $M(K) := g_j(K_1, \dots, K_{\mu_j - 1}, K, K_{\mu_j + 1}, \dots, K_n)$ and requests an input $D =: (M, h, (P, K'))$ with that M.

[6] Note that in this *function definition*, K is the argument while the K_{μ_i} are hardwired. In particular, B does not need to know the actual value of K for this step.

(Note that D may be D^R or D^F.) Then B sets the new key $K_{\mu_j} := K'$ and gives $(h, M \oplus P)$ as the ciphertext to A.
- For all further encryptions queries, B computes the real ciphertext using the keys $K_1, \ldots, K_n$ produced in the preceding steps.
- Finally, B outputs the output of A.

It is now easy to verify that if B gets D^R as input, B simulates the Game $j-1$, and if B gets D^F as input, B simulates the Game j. Hence

$$\begin{aligned}
&\mathsf{Pr}\left[B(D^R) = 1\right] - \mathsf{Pr}\left[B(D^F) = 1\right] \\
&= \frac{1}{p(k)} \sum_{j=1}^{p(k)} \mathsf{Pr}\left[A = 1 \text{ in Game } j-1\right] - \frac{1}{p(k)} \sum_{j=1}^{p(k)} \mathsf{Pr}\left[A = 1 \text{ in Game } j\right] \\
&= \frac{1}{p(k)} \big(\mathsf{Pr}\left[A = 1 \text{ in Game } 0\right] - \mathsf{Pr}\left[A = 1 \text{ in Game } p(k)\right]\big).
\end{aligned}$$

The right hand side is not negligible by assumption, thus the right hand side is not negligible either. This contradicts (4) and thus concludes the proof.

5.3 The usefulness of stateful KDM security

In a sense, strong stateful KDM security is "just as good" as standard KDM security. Arbitrarily large messages (in particular keys) can be encrypted by splitting up the message into parts and encrypting each part individually. The key-depencies of the message parts can be preserved, since the dependencies across states (i.e., dependencies on earlier keys) are allowed. This technique is generally *not* possible with weak stateful KDM security. We know of no weakly stateful KDM secure scheme with which one could securely encrypt one's own key (let alone construct key cycles).

But despite the drawbacks of weak stateful KDM security, we believe that this notion is still useful: first, it serves as a stepping stone towards achieving strong stateful KDM security (or even stateless KDM security). Second, in certain applications, weak stateful KDM security might be sufficient. Imagine, e.g., a setting in which the encrypted message contains side-channel information (like, say, internal measurements from the encryption device) on the internal state/secret key. If we assume that the old state is erased after encryption, the side-channel information only refers to the *current* internal state, and weak stateful KDM security is enough to provide message secrecy. Third, weak stateful KDM security provides an alternative assumption to the assumption of absence of key cycles in the formal protocol analysis setting. Instead of assuming the absence of key cycles (this assumption may not make sense in a scheme in which the key space is larger than the message space), we can assume that the encrypted terms depend only on the current internal state of the encryption algorithm. This assumption is still a strengthening of standard IND-CPA security and makes sense, since the encryption algorithm is only used to encrypt.

References

1. Martín Abadi and Phillip Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). *Journal of Cryptology*, 15(2):103–127, 2002. Extended version online available at `http://www.cse.ucsc.edu/~abadi/Papers/equiv.ps`.

2. Pedro Adão, Gergei Bana, Jonathan Herzog, and Andre Scedrov. Soundness of formal encryption in the presence of key-cycles. In Sabrina De Capitani di Vimercati, Paul F. Syverson, and Dieter Gollmann, editors, *Computer Security, Proceedings of ESORICS 2005*, number 3679 in Lecture Notes in Computer Science, pages 374–396. Springer-Verlag, 2005. Online available at `http://wslc.math.ist.utl.pt/ftp/pub/AdaoPM/05-ABHS-cycles.ps`.
3. Michael Backes, Birgit Pfitzmann, and Andre Scedrov. Key-dependent message security under active attacks – BRSIM/UC-soundness of symbolic encryption with key cycles. In *20th IEEE Computer Security Foundations Symposium, Proceedings of CSF 2007*, pages 112–124. IEEE Computer Society, 2007. Extended version online available at `http://eprint.iacr.org/2005/421.ps`.
4. Michael Backes, Birgit Pfitzmann, and Michael Waidner. A composable cryptographic library with nested operations. In *10th ACM Conference on Computer and Communications Security, Proceedings of CCS 2003*, pages 220–230. ACM Press, 2003. Extended abstract, extended version online available at `http://eprint.iacr.org/2003/015.ps`.
5. Mihir Bellare, Anand Desai, Eron Jokipii, and Phillip Rogaway. A concrete security treatment of symmetric encryption. In *38th Annual Symposium on Foundations of Computer Science, Proceedings of FOCS 1997*, pages 394–403. IEEE Computer Society, 1997. Extended version online available at `http://www.cs.ucsd.edu/users/mihir/papers/sym-enc.ps`.
6. Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In Hugo Krawczyk, editor, *Advances in Cryptology, Proceedings of CRYPTO '98*, number 1462 in Lecture Notes in Computer Science, pages 26–45. Springer-Verlag, 1998. Extended version online available at `http://eprint.iacr.org/1998/021.ps`.
7. John Black, Phillip Rogaway, and Thomas Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In Kaisa Nyberg and Howard M. Heys, editors, *Selected Areas in Cryptography, Proceedings of SAC 2002*, number 2595 in Lecture Notes in Computer Science, pages 62–75. Springer-Verlag, 2003. Online available at `http://eprint.iacr.org/2002/100.ps`.
8. Jan Camenisch and Anna Lysyanskaya. Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. In Birgit Pfitzmann, editor, *Advances in Cryptology, Proceedings of EUROCRYPT 2001*, number 2045 in Lecture Notes in Computer Science, pages 93–118. Springer-Verlag, 2001. Extended version online available at `http://www.zurich.ibm.com/%7Ejca/papers/eprint.pdf`.
9. Ran Canetti, Shai Halevi, and Jonathan Katz. Adaptively-secure, non-interactive public-key encryption. In Joe Kilian, editor, *Theory of Cryptography, Proceedings of TCC 2005*, number 3378 in Lecture Notes in Computer Science, pages 150–168. Springer-Verlag, 2005. Full version online available at `http://eprint.iacr.org/2004/317.ps`.
10. Ran Canetti, Hugo Krawczyk, and Jesper B. Nielsen. Relaxing chosen-ciphertext security. In Dan Boneh, editor, *Advances in Cryptology, Proceedings of CRYPTO 2003*, number 2729 in Lecture Notes in Computer Science, pages 565–582. Springer-Verlag, 2003. Full version online available at `http://eprint.iacr.org/2003/174.ps`.
11. Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Hugo Krawczyk, editor, *Advances in Cryptology, Proceedings of CRYPTO '98*, number 1462 in Lecture Notes in Computer Science, pages 13–25. Springer-Verlag, 1998. Online available at `http://eprint.iacr.org/1998/006.ps`.
12. Ivan Bjerre Damgård. A design principle for hash functions. In Gilles Brassard, editor, *Advances in Cryptology, Proceedings of CRYPTO '89*, number 435 in Lecture Notes in Computer Science, pages 416–427. Springer-Verlag, 1990.
13. Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography. In *Twenty-Third Annual ACM Symposium on Theory of Computing, Proceedings of STOC 1991*, pages 542–552. ACM Press, 1991. Extended abstract, full version online available at `http://www.wisdom.weizmann.ac.il/~naor/PAPERS/nmc.ps`.
14. Danny Dolev and Andrew C. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, 1983.

15. Cynthia Dwork, Moni Naor, Omer Reingold, and Larry Stockmeyer. Magic functions. *Journal of the ACM*, 50(6):852–921, 2003. Extended version online available at `http://www.wisdom.weizmann.ac.il/~naor/PAPERS/magic.ps`.
16. Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, April 1984.
17. Shai Halevi and Hugo Krawczyk. Security under key-dependent inputs. In *14th ACM Conference on Computer and Communications Security, Proceedings of CCS 2007*. ACM Press, 2007. Full version online available at `http://eprint.iacr.org/2007/315.pdf`.
18. Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999. Full version online available at `http://www.icsi.berkeley.edu/~luby/PAPERS/hill.ps`.
19. Dennis Hofheinz, Jörn Müller-Quade, and Rainer Steinwandt. On modeling IND-CCA security in cryptographic protocols. *Tatra Mountains Mathematical Publications*, 33:83–97, 2006.
20. Daniele Micciancio and Bogdan Warinschi. Soundness of formal encryption in the presence of active adversaries. In Moni Naor, editor, *Theory of Cryptography, Proceedings of TCC 2004*, number 2951 in Lecture Notes in Computer Science, pages 133–151. Springer-Verlag, 2004. Full version online available at `http://www.cs.ucsd.edu/~bogdan/ps/sfa.ps`.
21. Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Joan Feigenbaum, editor, *Advances in Cryptology, Proceedings of CRYPTO '91*, number 576 in Lecture Notes in Computer Science, pages 433–444. Springer-Verlag, 1992. Online available at `http://research.microsoft.com/crypto/dansimon/me.htm`.