Key-Dependent Message Security: Generic Amplification and Completeness

Benny Applebaum*

March 31, 2013

Abstract

Key-dependent message (KDM) secure encryption schemes provide secrecy even when the attacker sees encryptions of messages related to the secret-key sk. Namely, the scheme should remain secure even when messages of the form $f(\mathsf{sk})$ are encrypted, where f is taken from some function class $\mathcal{F}$. A KDM *amplification* procedure takes an encryption scheme which satisfies $\mathcal{F}$-KDM security, and boosts it into a $\mathcal{G}$-KDM secure scheme, where the function class $\mathcal{G}$ should be richer than $\mathcal{F}$. It was recently shown by Brakerski et al. (TCC 2011) and Barak et al. (EUROCRYPT 2010), that a strong form of amplification is possible, provided that the underlying encryption scheme satisfies some special additional properties.

In this work, we prove the first *generic* KDM amplification theorem which relies solely on the KDM security of the underlying scheme without making any other assumptions. Specifically, we show that an elementary form of KDM security against functions in which each output bit either copies or flips a single bit of the key (aka *projections*) can be amplified into KDM security with respect to any function family that can be computed in arbitrary fixed polynomial-time. Furthermore, our amplification theorem and its proof are insensitive to the exact setting of KDM security, and they hold in the presence of multiple-keys and in the symmetric-key/public-key and the CPA/CCA cases. As a result, we can amplify the security of most known KDM constructions, including ones that could not be amplified before.

Finally, we study the minimal conditions under which full-KDM security (with respect to all functions) can be achieved. We show that under strong notion of KDM security, the existence of fully-homomorphic encryption which allows to encrypt the secret-key (i.e., "cyclic-secure") is not only sufficient for full-KDM security, as shown by Barak et al., but also necessary. On the other hand, we observe that for standard KDM security, this condition can be relaxed by adopting Gentry's bootstrapping technique (STOC 2009) to the KDM setting.

1 Introduction

The study of secure encryption scheme is perhaps the most central subject in cryptography. Since the introduction of semantic security [28] through the formulation of CCA-security [35, 37, 21] and more, modern cryptography has successfully developed increasingly stronger notions of security

*School of Electrical Engineering, Tel-Aviv University, benny.applebaum@gmail.com. Work done in part while a postdoc at the Weizmann Institute of Science, supported by Koshland and Alon Fellowships, by the Israel Science Foundation (grant No. 1155/11), and by the Check Point Institute for Information Security. An extended abstract of this paper appears in the proceedings of Eurocrypt'11.

providing secrecy in highly adversarial settings. Still, all these strong notions of security guarantee secrecy only as long as the encrypted messages are independent of the secret key. This limitation dates back to the seminal work of Goldwasser and Micali [28] who observed that semantic security may not hold if the adversary gets to see an encryption of the secret key. For many years, such usage scenarios were considered as "security bugs" that should be prevented by system designers.

A decade ago, the assumption of independency between the secret key and the encrypted data was challenged by Camenisch and Lysyanskaya [19] and Black et al. [12]. Specifically, Camenisch and Lysyanskaya considered schemes that remain secure under a "key cycle" usage, where we have t keys organized in a cycle and each key is encrypted under its left neighbor. A generalization of this notion, called *key-dependent message* (KDM) security, was suggested by Black et al. Informally, an encryption is $\mathsf{KDM}^{(t)}$ secure with respect to a function class $\mathcal{F}$ if security holds even when the adversary can ask for an encryption of the message $M = f(\mathsf{sk}_1, \ldots, \mathsf{sk}_t)$ under the i-th public-key, where $\mathsf{sk}_1, \ldots, \mathsf{sk}_t$ are the secret keys present in the system and f is an arbitrary function in $\mathcal{F}$. This notion of security implies cyclic-security if $\mathcal{F}$ is expressive enough (e.g., contains all "selector" functions), and it becomes stronger when the function class $\mathcal{F}$ grows. Hence, one would like to achieve KDM security while making the function class $\mathcal{F}$ as large as possible.

The notion of KDM security was extensively studied in the past few years in several flavors including the symmetric/public-key and the CPA/CCA settings [19, 12, 30, 10, 13, 17, 8, 31, 29, 5, 15, 2, 11, 14]. These works were motivated by the fundamental nature of the question as well as by concrete applications including encrypted storage systems (e.g., BitLocker [13]), anonymous credentials [19], and realization of security proofs at the framework of axiomatic security [1, 12, 3]. (See [13] for more motivations and details.)

Although much is known today about KDM security both on the positive and negative sides, it is still unclear whether a standard encryption scheme can be transformed into a scheme which provides $\mathsf{KDM}^{(t)}$ security, even with respect to a single key (i.e., $t = 1$) and simple non-trivial function families (e.g., selectors).[1] Hence, it is natural to move forward and explore the possibility of building strong KDM security given a weak form of KDM security as a primitive. This makes sense as today, following the seminal work of Boneh et al. [13] and its follow-ups [17, 5, 14], it is known that a basic form of KDM security (with respect to the family of "affine functions") can be achieved in several settings under various concrete cryptographic assumptions. Therefore, we ask:

> Is there a *generic* transformation which *amplifies* KDM security from a weak family of functions $\mathcal{F}$ to a larger family of functions $\mathcal{G}$?

The question of KDM amplification was recently addressed by Brakerski et al. [15] and Barak et al. [11], who made an important progress by showing how to amplify the KDM security of several existing schemes. While the resulting amplification procedures are relatively powerful (there is a considerable difference between $\mathcal{F}$ and $\mathcal{G}$), they fall short of providing full *generality* as they strongly rely on additional properties of the underlying scheme (i.e., *simulatable*-KDM security and *entropic*-KDM security – to be defined later). As a concrete example, it is unknown how to use any of these techniques to amplify the KDM-security of the symmetric-key encryption scheme of [5] which is based on the Learning Parity With Noise (LPN) assumption. (See Section 1.3 for more details about these works and their relation to our approach.)

[1] Known impossibility results [29, 11] only hold with respect to sufficiently rich families of functions (e.g., capable of computing poly(k)-independent hash functions, or pseudorandom functions).

1.1 Our Results

We give an affirmative answer to the above question by providing the first generic KDM amplification procedure. In particular, we consider the *projection* function class of all functions $f : (\mathsf{sk}_1, \ldots, \mathsf{sk}_t) \mapsto v$ in which each output bit depends on (at most) a single bit of the input. Namely, each output bit v_j is either fixed to a constant or copies/flips an original bit of one of the keys. We show that this elementary function family is *complete* in the following sense:

Theorem 1.1 (Completeness of projections, Informal)**.** *Let $\mathcal{G}$ be any function family which can be computed in some fixed polynomial time. Then, any encryption scheme which satisfies $\mathsf{KDM}^{(t)}$ security with respect to projections can be transformed into a new encryption scheme which is $\mathsf{KDM}^{(t)}$-secure with respect to $\mathcal{G}$.*

Generality. Theorem 1.1 assumes nothing but KDM security regarding the underlying scheme. Furthermore, the theorem (and its surprisingly simple proof) is insensitive to the exact setting of KDM security: it holds for any number of keys t, and in both symmetric-key/public-key and CPA/CCA settings. In all these cases, the new scheme is proven to be secure exactly in the same setting as the original scheme. This allows us, for example, to amplify the security of the affine-KDM secure scheme of [5], and obtain the first symmetric-key encryption scheme with strong KDM security based on the LPN assumption.

Extensions. Theorem 1.1 can be further strengthened as follows. First, we can achieve *length-dependent* KDM security [11], which means that the target family $\mathcal{G}$ can be taken to be the family of all polynomial-size circuits whose size grows with their input and output lengths via a fixed polynomial rate (e.g., the circuit size is quadratic in the input and output lengths). This family is very powerful and it was shown to be rich enough for most known applications of KDM security [11].[2] (See Section 2 for details.) In addition, in the case of CPA security (both in the public-key and symmetric-key settings), we can weaken the requirement from the underlying scheme and ask for KDM security with respect to projections with a *single output*: namely, all Boolean functions $f(\mathsf{sk}_1, \ldots, \mathsf{sk}_t) \mapsto b$ which output a single bit of one of the keys, or its negation. This can be extended to the CCA setting via the transformations of [10, 17] (though in the public-key setting one has to employ, in addition, non-interactive zero-knowledge proofs).

The relaxation to single-output projections also enables a liberal interface to which we can easily plug previous constructions. For example, one can instantiate our reduction with schemes that enjoy KDM security with respect to affine functions, while almost ignoring technical details such as the underlying field and its representation. (These details required some effort in previous works. See the appendices in [15, 11, 14].) This, together with the simple proof of our main theorem, allows to simplify the proofs of [11, 14] for the existence of length-dependent KDM secure encryption scheme under the Decisional Diffie-Hellman (DDH) assumption [13], the Learning With Errors assumptions (LWE) [5], and the Quadratic Residuosity (QR) and Paillier's Decisional Composite Residuosity (DCR) assumptions [14].

Given this completeness theorem, the current status of KDM security resembles the status of other "complete" primitives in cryptography such as one-way functions or oblivious transfer [36, 22]:

[2] Most of the statements in [11] refer to the slightly weaker notion of *Bounded KDM security* in which the circuit size grows only as a function of the input via a fixed polynomial rate. However, as observed in [11, Sec. 6] their construction actually satisfies the stronger definition of *length-dependent* KDM security.

We do not know how to build these primitives from generic weaker assumptions, however, any instantiation of them suffices for an entire world of applications (i.e., all symmetric-key primitives in the case of one-way functions, and generic secure-computation in the case of oblivious transfer, cf. [26, 27]).

Beyond length-dependent security. Although length-dependent KDM security seems to suffice for most applications, one can strive for an even stronger notion of security in which the KDM function class contains all functions (or equivalently all functions computable by circuits of *arbitrary* polynomial size). It is somewhat likely that any length-dependent secure scheme actually achieves *full-KDM* security (see the discussion in [11]). Still, one may want to construct such a scheme in a provably secure way. As a basic feasibility result, it was shown in [11] that any fully homomorphic encryption scheme [23] which allows to encrypt the secret-key (i.e., "cyclic-secure") is also full-KDM secure. Unfortunately, despite the recent progress in the study of FHEs (cf. [38] and references there) it is still unknown how to construct cyclic-secure FHEs under standard assumptions.[3] Hence, one may ask whether it is possible to relax this requirement and achieve full-KDM security under weaker assumptions.

We make two simple observations regarding this question. First, we consider the case of simulatable KDM security [11], in which it should be possible to simulate an encryption of $f(\mathsf{sk})$ given only the corresponding public-key in a way that remains indistinguishable even to someone who knows the secret-key. We show that in this setting the two notions: circular-secure FHE and full-KDM are equivalent. Hence, achieving full-KDM security under a relaxed assumption requires to use non-simulatable constructions.

Our second observation asserts that the bootstrapping technique of Gentry [23] can be used in the KDM setting as well (even for the case of non-simulatable constructions). That is, if one can construct an encryption scheme which guarantees KDM security with respect to circuits whose depth is only slightly larger than the depth of the decryption algorithm, then this scheme is actually fully KDM secure. Unfortunately, all known amplification techniques [11, 15] including the ones in this paper, amplify KDM security at the cost of making the decryption algorithm "deeper". Still, we view this observation as an interesting direction for future research.

1.2 Our Techniques

To formalize the question of KDM amplification, we define the notion of *reduction* between KDM function families $\mathcal{G} \leq_{\mathsf{KDM}} \mathcal{F}$ which means that any scheme that provides KDM security with respect to $\mathcal{F}$ can be transformed via a fully black-box reduction to a new scheme that satisfies KDM security with respect to $\mathcal{G}$.[4] We describe a novel way to derive such KDM reductions based on the machinery of *randomized encoding* of functions [33, 7]. Before we explain this notion, let us start with the simpler case of *deterministic encoding*.

[3] One can use standard assumptions to construct a *leveled* homomorphic-encryption (LHE) which supports homomorphic operations up to a-priori known bounded depth (e.g., [16, 24]). However, the only known transformation from LHE to FHE requires an additional cyclic security assumption. This further motivates the study of KDM security.

[4] The term *fully black-box reduction* means (as usual in cryptography) that the new scheme makes only black-box use of the original scheme, and that the security proof of the construction is also black-box in the sense that an adversary breaking the new scheme can be used as an oracle in order to break the underlying scheme. In contrast, our security proof makes a *non black-box* use of the KDM family. See later discussion.

Say that a function f deterministically encodes a function g if for every x the output of $f(x)$ "encodes" the output of $g(x)$ in the sense that $g(x)$ can be efficiently computed based on $f(x)$ and vice versa. That is, there are two efficiently computable mappings S and R such that $S(g(x)) = f(x)$, and $R(f(x)) = g(x)$. Suppose that we are given a scheme which provides KDM security with respect to the encoding f, and we would like to immunize it against the function g. This can be easily achieved by modifying the encryption scheme as follows: to encrypt a message M we first translate it into the f-encoding by computing $S(M)$, and then encrypt the result under the original encryption scheme. Decryption is done by applying the original decryption algorithm, and then applying the recovery algorithm R to translate the result back to its original form. Observe that an encryption of $g(\mathsf{sk})$ in the new scheme is the same as an encryption of $S(g(\mathsf{sk})) = f(\mathsf{sk})$ under the original scheme. Hence, the KDM security of the new scheme with respect to g reduces to the KDM security of the original scheme with respect to f.

This simple idea provides a direct reduction with very nice structure: any KDM query for the new scheme is translated into a single KDM query for the original scheme. This simple single-query-to-single-query translation leads to high level of generality: the transformation is insensitive to the exact KDM setting (symmetric-key/public-key and CPA/CCA), to the number of keys, and it can be used with respect to large function families $\mathcal{G}$ and $\mathcal{F}$ as long as every function in $\mathcal{G}$ is encoded by some function in $\mathcal{F}$ via a pair of universal mappings S and R. On the down side, one may complain that security was not really *amplified*, as the function g and its encoding f are essentially equivalent. It turns out that this drawback can be easily fixed by letting f be a *randomized* encoding of g.

In the case of randomized encoding (RE), the function $f(x; r)$ depends not only on x but also on an additional random input r. For every fixed x, the output of $f(x; r)$ is now viewed as a probability distribution (induced by a random choice of r) which should encode the value of $g(x)$. Namely, there are two efficiently computable randomized mappings S and R such that for every x: (1) the distribution $S(g(x))$ is indistinguishable from $f(x; r)$, and (2) with high probability over the choice of r (or even with probability one) $R(f(x; r)) = g(x)$. One can view these conditions as saying that $g(x)$ is encoded by a *collection* of functions $\{f_r(x)\}_r$, where $f_r(x) = f(x; r)$.

Now suppose that our scheme is KDM secure with respect to the family $\{f_r(x)\}_r$, then we can apply the above approach and get a scheme which satisfies KDM security with respect to g. The only difference is that now the message preprocessing step is randomized: To encrypt a message M first encode it by the randomized mapping $S(M)$, and then use the original encryption function. The security reduction is essentially the same except that a KDM query for g in the new scheme is emulated by an old KDM query for a *randomly chosen* function f_r. This idea can be easily extended to the case where all functions in $\mathcal{G}$ are encoded by functions in $\mathcal{F}$:

Theorem 1.2 (Informal)**.** *If $\mathcal{F}$ is an RE of $\mathcal{G}$, then $\mathcal{G} \leq_{\mathsf{KDM}} \mathcal{F}$.*

The crux of this theorem, is that, unlike deterministic encoding, randomized encoding can represent complicated functions by collections of very simple functions [33, 34, 7, 6]. Specifically, by combining the above theorem with the REs of [6], which, in turn, are based on Yao's garbled circuit [39], we obtain our main results (Theorem 1.1).

1.3 Comparison with BGK and BHHI

Our techniques are inspired by both [15] (BGK) and [11] (BHHI). We believe that our approach inherits the positive features of each of these works, and sheds new light on the way they relate to

each other. Let us review the main ideas behind these constructions and explain how they compare to our solution.

1.3.1 The BGK reduction

The starting point in [15] is an encryption scheme which satisfies entropic KDM security with respect to $\mathcal{F}$. Roughly speaking, this means that KDM security should hold not only when sk is chosen uniformly from the key space $\mathcal{K} = \{0,1\}^k$ but also when it is chosen uniformly from a smaller domain $\mathcal{K}'$, e.g., $\mathcal{K}' = \{0,1\}^{k^\varepsilon}$. By relying on this notion, BGK shows that for every efficiently computable injective mapping $\alpha : \mathcal{K}' \to \mathcal{K}$, one can amplify security from $\mathcal{F}$ to the class $\mathcal{F} \circ \alpha$, i.e., with respect to functions $f(\alpha(\mathsf{sk}))$ for every $f \in \mathcal{F}$. The idea is to choose the key sk' from $\mathcal{K}'$ and employ the original scheme with the key $\mathsf{sk} = \alpha(\mathsf{sk}')$. This allows to translate a KDM query $f(\alpha(\mathsf{sk}'))$ for the new scheme into an entropic-KDM query $f(\mathsf{sk})$ for the old scheme.

The deterministic encoding (DE) approach is inspired by the BGK approach, and can be seen as a complementary solution. BGK extends a function $f : \mathcal{K} \to \mathcal{M}$ to $f \circ \alpha : \mathcal{K}' \to \mathcal{M}$ by shrinking the key space (from $\mathcal{K}$ to $\mathcal{K}'$), whereas in the DE approach $f : \mathcal{K} \to \mathcal{M}$ is extended to $R \circ f : \mathcal{K} \to \mathcal{M}'$ by padding messages which effectively shrinks the message space (from $\mathcal{M}$ to $\mathcal{M}' = R(\mathcal{M})$).

As a result BGK enjoys a similar attractive security reduction with single-query-to-single-query translation. This leads to flexibility with respect to the KDM *setting*. Indeed, although the BGK approach is not fully general due to its use of entropic-KDM security (a notion which seems stronger than standard KDM security), it immediately generalizes to the CCA and the symmetric-key settings, as long as the underlying scheme provides entropic-KDM security.

It should be mentioned that in our approach the amplification is achieved by modifying the encryption algorithm, rather than the key-generation algorithm as in BGK. This minor difference turns to have a considerable effect. First, it allows to use fresh randomness in every application of the encryption algorithm, and so the linkage between functions in $\mathcal{G}$ to functions in $\mathcal{F}$ can be *randomized*. Indeed, this is exactly what allows us to exploit the power of randomized encoding. In contrast, the BGK approach tweaks the key-generation algorithm and so the relation between $\mathcal{G}$ to $\mathcal{F}$ is bounded to be deterministic. In addition, since our modification happens in the encryption (and decryption) phases, we can let the function class $\mathcal{G}$ grow not only with the security parameter but also with the size of the messages. This leads to the strong notion of length-dependent security, and in addition allows to achieve $\mathsf{KDM}^{(t)}$ where the number of keys t grows both with the message length and the security parameter.

In contrast, the family $\mathcal{G}$ of BGK cannot grow with the message length, and it can only contain a polynomial number of functions. This limitation prevents it from being used in applications which require KDM security with respect to larger functions classes (e.g., secure realization of symbolic protocols with axiomatic proofs of security). Furthermore, amplification for large number of keys can be achieved only at the expense of putting more restrictions on the underlying scheme (i.e., simulatable KDM security). On the other hand, assuming these additional properties, the BGK approach can get KDM security for concrete functions (e.g., constant degree polynomials) which involve an arbitrary unbounded number of keys t, whereas in our approach the arity of the KDM function is always bounded by some fixed predefined polynomial in the security parameter and message length.[5] Finally, it is important to mention that the BGK reduction treats $\mathcal{G}$ in a

[5] More precisely, our reduction limits the circuit size of the KDM function and therefore also its arity; However, it puts no restriction on the number of keys in the system. Hence, if our transformation is applied to a scheme which

black-box way, while the randomized encoding approach treats this class in a non-black-box way.

1.3.2 The BHHI reduction

The BHHI approach relies on a novel connection between homomorphic encryptions and KDM security. First, it is observed that in order to obtain KDM security with respect to $\mathcal{G}$ it suffices to construct a scheme which provides both cyclic-security (i.e., KDM security with respect to the identity function) and homomorphism with respect to a function family $\mathcal{G}$, i.e., it should be possible to convert a ciphertext $C = \mathsf{E}_{\mathsf{pk}}(M)$ into $C' = \mathsf{E}_{\mathsf{pk}}(g(M))$ for every $g \in \mathcal{G}$. Indeed, the homomorphism property can be used to convert a ciphertext $\mathsf{E}_{\mathsf{pk}}(\mathsf{sk})$ into the ciphertext $\mathsf{E}_{\mathsf{pk}}(g(\mathsf{sk}))$, and so cyclic-security is amplified to $\mathcal{G}$-KDM security.

BHHI construct such an encryption scheme by combining a two-party secure computation protocol with two messages (i.e., based on Yao's garbled circuit [39]) with a strong version of oblivious transfer which satisfies an additional *cyclic-security* property. The latter primitive is referred to as *targeted encryption* (TE). The basic idea is to view the homomorphic property as a secure-computation task in which the first party holds the message M and the second party holds the function g. The cyclic nature of the TE primitive allows to implement this homomorphism even when the input M is the secret-key. Finally, BHHI show that TE can be constructed based on affine-KDM secure encryption scheme which satisfies a strong notion of simulation: There exists a simulator which given the public-key pk can simulate a ciphertext $\mathsf{E}_{\mathsf{pk}}(g(\mathsf{sk}))$ in a way which is indistinguishable even for someone who holds the secret-key.

The BHHI construction seems conceptually different from our RE approach (i.e., homomorphism vs. encoding). Moreover, the construction itself is not only syntactically different, but also relies on different building blocks (e.g., TE). Still, the RE construction shares an important idea with BHHI: The use of secure-computation techniques. It is well known that REs are closely related to secure multiparty-computation (MPC) protocols [33], and, indeed, the role of REs in our reduction resembles the role of MPC in BHHI. In both solutions at some point the security reduction applies the RE/MPC to the function g in $\mathcal{G}$. Furthermore, both works achieve strong KDM security by instantiating the RE/MPC with Yao's garbled circuit (GC) — a tool which leads to both stand-alone RE construction [6] and, when equipped with an OT, to a two-party secure-computation protocol.

It should be emphasized, however, that the actual constructions differ in some important aspects. While we essentially encrypt the whole GC-based encoding under the underlying KDM encryption scheme, BHHI tweak the GC protocol with a cyclic-secure OT (i.e., TE). Pictorially, our underlying KDM-secure scheme "wraps" the GC encoding, whereas in BHHI the KDM-secure primitive is "planted inside" the GC protocol. This difference affects both generality and simplicity as follows.

First, BHHI are forced to implement a KDM-secure OT, a primitive which seems much stronger than standard KDM secure encryption schemes. For example, KDM-secure symmetric-key encryption schemes can be constructed at the presence of a random oracle [12] while OT protocols cannot [32].[6] Moreover, as we already mentioned, although TE can be based on several known

satisfies $\mathsf{KDM}^{(t)}$ security for arbitrary t's (as in [13, 5]), we obtain an encryption scheme which provides $\mathsf{KDM}^{(t)}$ security even when the number of keys t in the system is unbounded, as long as the size (and arity) of the KDM functions available to the adversary is bounded by some (predetermined) polynomial. See Remarks 2.2 and 3.8.

[6] It seems that a similar statement holds even for public-key KDM-secure schemes. See [12, 25].

affine-secure KDM schemes (i.e., ones which enable strong simulation), the LPN assumption (with constant error-rate) is a concrete example under which symmetric-key encryption scheme with KDM-security with respect to affine functions exist, yet OT is not known to exist. Furthermore, since BHHI send the garbled circuit in the clear, it is not hard to show that the resulting scheme is not CCA-secure even if the TE provides CCA security. Finally, the modification of the GC protocol leads to a relatively complicated security proof, which relies on non-standard properties of the GC (e.g., "Security against outsiders"), and requires non-trivial additional work in order to achieve $\mathsf{KDM}^{(t)}$ security with multiple keys (i.e., for large t).

2 KDM-Security

2.1 Definitions

Notation. For a positive integer $n \in \mathbb{N}$, let $[n]$ denote the set $\{1, \ldots, n\}$. A function $\varepsilon(k) : \mathbb{N} \to [0,1]$ is *negligible* if it tends to zero faster than $1/k^c$ for every constant $c > 0$. We let $\mathrm{neg}(k)$ denote an arbitrary negligible function in k (i.e., when we say that $f(k) < \mathrm{neg}(k)$ we mean that there exists a negligible function $\varepsilon(k)$ such that for every $k, f(k) < \varepsilon(k)$). The term *efficient* refers to probabilistic machines that run in polynomial time in the security parameter.

Encryption schemes (syntax). An encryption scheme consists of three efficient algorithms $(\mathsf{KG}, \mathsf{E}, \mathsf{D})$, where KG is a key generation algorithm which given a security parameter 1^k outputs a pair $(\mathsf{sk}, \mathsf{pk})$ of decryption and encryption keys; E is an encryption algorithm that takes a message $M \in \{0,1\}^*$ and an encryption key pk and outputs a ciphertext C; and D is a decryption algorithm that takes a ciphertext C and a decryption key sk and outputs a plaintext M'. We also assume that both algorithms take the security parameter 1^k as an additional input, but typically omit this dependency for simplicity. We emphasize that the time complexity of E and D is polynomial in $k + \ell$ where k is the security parameter and ℓ is the length of the message/ciphertext, respectively.

Encryption schemes should satisfy *correctness*, which requires that for each message $M \in \{0,1\}^*$

$$\Pr_{(\mathsf{sk},\mathsf{pk}) \xleftarrow{R} \mathsf{KG}(1^k)} [\mathsf{D}_{\mathsf{sk}}(\mathsf{E}_{\mathsf{pk}}(M)) \neq M] \leq \delta(k),$$

where the *decryption error* of the scheme $\delta(k)$ should be negligible, and the probability is taken over the randomness of KG, E and D. For security parameter k, let $\mathcal{K}_k$ denote the space from which decryption keys are chosen. We assume, without loss of generality, that the binary representation of elements from $\mathcal{K}_k$ is k bit long.

Following Goldreich [27], we note that the above definition captures both public-key and symmetric-key encryption schemes where the latter corresponds to the special case in which the decryption key sk and encryption key pk are equal. As we will see, the difference between the two settings will be part of the security definitions.

KDM ensembles. Let $t : \mathbb{N} \to \mathbb{N}$ be a function that determines the number of keys, and let $\ell : \mathbb{N} \to \mathbb{N}$ be a length function. A t-ary *KDM function ensemble* with output length ℓ is a collection of functions $\mathcal{F} = \left\{ f_{k,z} : \mathcal{K}_k^{t(k)} \to \{0,1\}^{\ell(|z|)} \right\}_{(k,z)}$ indexed by the security parameter k and an identification string z, where each function $f_{k,z}$ maps a tuple of $t(k)$ keys in $\mathcal{K}_k$ into a

- **Initialization.** The challenger randomly chooses a bit $b \xleftarrow{R} \{0,1\}$ and $t = t(k)$ key-pairs $(\mathsf{sk}_1, \mathsf{pk}_1) \ldots, (\mathsf{sk}_t, \mathsf{pk}_t)$ by invoking $\mathsf{KG}(1^k)$ for t times. The adversary $\mathcal{A}$ can send a "public-key" query and get to see all the encryption keys $(\mathsf{pk}_1, \ldots, \mathsf{pk}_t)$.
- **Queries.** The adversary $\mathcal{A}$ may adaptively make polynomially-many queries of the following types:
  - **Encryption queries** of the form (i, M) where $i \in [t]$ and $M \in \{0,1\}^*$. The challenger responds with $C \xleftarrow{R} \mathsf{E}(\mathsf{pk}_i, M)$ if $b = 1$, and $C \xleftarrow{R} \mathsf{E}(\mathsf{pk}_i, 0^{|M|})$ if $b = 0$.
  - **KDM queries** of the form (i, z) where $i \in [t]$ and $z \in \{0,1\}^*$. The challenger computes $M = f_{k,z}(\mathsf{sk}_1, \ldots, \mathsf{sk}_t)$ and responds with $C \xleftarrow{R} \mathsf{E}(\mathsf{pk}_i, M)$ if $b = 1$, and $C \xleftarrow{R} \mathsf{E}(\mathsf{pk}_i, 0^{|M|})$ if $b = 0$.
  - **Decryption queries** of the form (i, C) where $i \in [t]$ and the string C was not given as an answer of a previous encryption/KDM query. The challenger responds with $M = \mathsf{D}_{\mathsf{sk}_i}(C)$ regardless of the value of b.
- **Final phase.** The adversary outputs a bit $b' \in \{0,1\}$ and wins if $b = b'$.

Figure 1: The $\mathcal{F}$-KDM game is defined with respect to the function ensemble $\mathcal{F} = \{f_{k,z}\}$ and is indexed by the security parameter k. The presence (resp., absence) of public-key query captures the public-key (resp., symmetric-key) setting.

message of length $\ell(|z|)$.[7] By default, the index z represents the circuit that computes the function $f_{k,z}$. We sometime abuse notation and identify $\mathcal{F}$ with the evaluation algorithm of the ensemble which maps $1^k, z$ and $\vec{\mathsf{sk}} \in \mathcal{K}_k^{t(k)}$ to $f_{k,z}(\vec{\mathsf{sk}})$. By convention, if $f_{k,z}$ is not in the collection we assume that $\mathcal{F}(1^k, z, \vec{\mathsf{sk}}) = 0$. The ensemble is *efficiently computable* if the time-complexity of the evaluation algorithm $\mathcal{F}$ is polynomial in the security parameter k. A weaker form of efficiency (for which our results also apply) allows the complexity of $\mathcal{F}$ to be polynomial in the security parameter and the output length, namely, to be bounded by $(\ell(|z|) + k)^c$ for some constant c. In this case we say that $\mathcal{F}$ is *output efficiently computable*. Output-efficient ensembles are strictly richer than efficient ensembles.

The KDM game. An $\mathcal{F}$-KDM Chosen-Ciphertext Attack (CCA) in the public-key setting is defined in Fig. 1 as a game that takes place between a challenger and an adversary $\mathcal{A}$. The *advantage* of $\mathcal{A}$ when attacking a scheme $\mathcal{E}$ is $\alpha(k) = \Pr[\mathcal{A} \text{ wins the KDM game}] - \frac{1}{2}$.

By restricting the power of the adversary in the KDM game (Fig. 1) we get other KDM settings. Specifically, the symmetric-key setting corresponds to adversaries of type sym who do not ask public-key queries, and the CPA setting corresponds to adversaries of type CPA who do not make decryption queries. Hence, we can classify KDM adversaries into one of the following four *types*: $(\mathsf{pub}, \mathsf{CCA}), (\mathsf{pub}, \mathsf{CPA}), (\mathsf{sym}, \mathsf{CCA})$, and $(\mathsf{sym}, \mathsf{CPA})$. An adversary of type T that conducts an $\mathcal{F}$-KDM attack is denoted as $(T, \mathcal{F})$-adversary.

[7] One could let ℓ depend on z itself, and not only on its length $|z|$. We prefer the current formulation for simplicity.

Definition 2.1. (KDM-secure encryption) *Let T be a type, and $\mathcal{F}$ be a function ensemble. An encryption scheme is* $(T, \mathcal{F})$-KDM secure *if every efficient $(T, \mathcal{F})$ adversary has at most negligible advantage when attacking the scheme.*

Remark 2.2 (Refined arity)**.** *Following previous works, we define KDM security with a single parameter t which denotes both the number of keys in the system and the arity of the KDM ensemble $\mathcal{F}$. We note that it is possible to use a more refined two-parameter definition, in which the arity of $\mathcal{F}$ is t and the total number of the keys that participate in the game is $\tau \geq t$. For example, imagine a scheme which remains secure when the KDM game is initialized with large (or even unbounded) number of keys, but the adversary is allowed to employ KDM functions which are applied to any* pair *of keys. This refined notion can be formalized by augmenting a KDM query (i, z) with an ordered t-size subset $S \subset [\tau]$, meaning that f_z should be applied to $(\mathsf{sk}_i)_{i \in S}$. For simplicity, we use a single parameter definition which assumes that τ, the number of keys in the system, is equal to t, the arity of the KDM ensemble. We note that some of our results hold even when τ is unbounded. (See Remark 3.8.)*

2.2 Examples of KDM ensembles

We consider several examples of t-ary KDM ensembles $\mathcal{F} = \{\mathcal{F}_k\}$.

Constants, selectors, and projections. If $\mathcal{F}_k$ contains all constant functions $\{f_M : (\mathsf{sk}_1, \ldots, \mathsf{sk}_t) \mapsto M\}_M$, then, as observed in [13], KDM queries are equivalent to standard encryption queries and KDM security is nothing but standard security (with respect to the type T). If the ensemble $\mathcal{F}_k$ contains all selector functions $\{f_j : (\mathsf{sk}_1, \ldots, \mathsf{sk}_t) \mapsto \mathsf{sk}_j\}_{j \in [t]}$, we get the notion of *clique* security [13] (which is stronger than *circular security* [19]), that is, the scheme is secure even if the adversary sees encryptions of the form $\mathsf{E}_{\mathsf{pk}_i}(\mathsf{sk}_j)$ for every $i, j \in [t]$. Another elementary class that slightly generalizes the previous ones is the class of all functions $f : \vec{\mathsf{sk}} \mapsto v$ in which each output bit depends on (at most) a single bit of the input $\vec{\mathsf{sk}} = (\mathsf{sk}_1, \ldots, \mathsf{sk}_t)$. Namely, the j-th output bit v_j is either fixed to a constant or copies/flips an original bit of one of the keys, i.e., $v_j \in \{0, 1, \mathsf{sk}_{i,q}, 1 - \mathsf{sk}_{i,q}\}$, where $\mathsf{sk}_{i,q}$ is the q-th bit of the i-th secret key. We refer to this class as the class of *projections* and let $\Pi^t_{k,\ell}$ denote the restriction of this class to functions of input length kt and output length $\ell(k)$. Projections is a proper subclass of the class of affine functions $L : \mathbb{F}_2^{kt} \to \mathbb{F}_2^{\ell(k)}$. Observe that all the above classes are efficiently computable. We also consider the class of projections of unbounded polynomial length $\Pi^t_k = \bigcup_{a \in \mathbb{N}} \Pi^t_{k,k^a}$ which is output-efficiently computable. The constructions of [13, 5, 14] (or variants of them) achieve $(\mathsf{pub}, \mathsf{CPA})$-KDM security with respect to Π^t_k for every polynomial $t(k)$.[8]

Polynomial-size circuits [11]. For polynomials $p(\cdot)$ and $\ell(\cdot)$, let $\mathcal{C}^t_{k,\ell,p}$ denote the class of all Boolean circuits $C : \{0,1\}^{kt} \to \{0,1\}^{\ell(k)}$ of size at most $p(k + \ell(k))$. For example, if $t = 1$ and ℓ, p are quadratic we get the family of all circuits $C : \{0,1\}^k \to \{0,1\}^{k^2}$ of size $(k + k^2)^2 \approx k^4$. It is not hard to see that $\mathcal{C}^t_{k,\ell,p}$ is efficiently computable as it can be computed by an efficient universal

[8] More precisely, [13, 5] present a *single* construction which works for every polynomial $t(k)$, while [14] provide for every polynomial t a *different* construction.

algorithm F which given $\vec{\mathsf{sk}}$ and a circuit C of size $p(k+\ell(k))$ evaluates $C(\vec{\mathsf{sk}})$ in time $\mathrm{poly}(kt)$. Security with respect to the class $\mathcal{C}^t_{k,\ell,p}$ is denoted by (p,ℓ)-*bounded circuit-size* KDM security.

We also consider a stronger variant of this notion as follows. A scheme is *p-length-dependent* KDM secure if it is (p,ℓ)-*bounded circuit-size* KDM secure for *every* polynomial $\ell(\cdot)$. Equivalently, it is KDM secure with respect to the class $\mathcal{C}^t_{k,p} = \bigcup_{a\in\mathbb{N}} \mathcal{C}^t_{k,k^a,p}$. For example, if $t=1$ and p is quadratic, we get security with respect to every circuit C whose size is quadratic in the output length. Although $\mathcal{C}^t_{k,p}$ is not efficiently computable, it is output efficiently computable as it can be evaluated in time $\mathrm{poly}(k,\ell)$.

Remark 2.3 (length-dependent security vs. full security)**.** *Full-KDM security corresponds to the case where $\mathcal{F}$ is the class of all functions. Recall that adversaries are assumed to be efficient, and each KDM query f_z is assumed to be specified via a description z of the circuit that computes f_z. Under these conventions, adversaries are always restricted to KDM functions which are polynomial-time computable (with respect to an arbitrary polynomial), and so, a scheme which is p-length-dependent secure with respect to* every *polynomial $p(\cdot)$ is also fully secure. We note that in some scenarios p-length-dependent security, say for quadratic p, may be considered to be almost as powerful as full KDM security. Indeed, length-dependent security allows the adversary to use larger circuits by encrypting longer messages. Therefore, although quadratic-length dependant scheme does not guarantee security when the adversary sees $\mathsf{E}_{\mathsf{pk}}(f(\mathsf{sk}))$ for a function f of complexity, say $O(k^3)$, we can prove security under a similar attack in which f is replaced with a padded version $f'(\mathsf{sk}) = (f(\mathsf{sk}), 0^{k^3})$. It seems that, at least in some scenarios, security against the latter attack is as useful as the former. Furthermore, in [11] it was shown that p length-dependent security (say for quadratic p) is sufficient for axiomatic-security applications (i.e., it gives the ability to securely instantiate symbolic protocols with axiomatic proofs of security).*

Remark 2.4 (The role of the arity t)**.** *Clearly, $\mathcal{F}$-KDM security becomes stronger when the arity t grows. At one extreme, one may consider a single encryption scheme which satisfies $\mathcal{F}$-KDM security for an arbitrary polynomial $t(k)$, and at the other extreme one may consider the case of $t=1$, which is still non-trivial even for projection functions.*

2.3 Encoding KDM ensembles

Intuitively, a randomized encoding of a function $g(x)$ is a randomized mapping $f(x;r)$ whose output distribution (which is induced by a random choice of r) depends only on the output of g. We formalize this intuition via the notion of *computationally private randomized encoding* of [6], while adopting the original definition from a non-uniform adversarial setting to the uniform setting (i.e., adversaries are modeled by probabilistic polynomial-time Turing machines), and tailoring it to the case of KDM ensembles.

Definition 2.5 (Randomized encoding)**.** *Let $G : 1^k \times \{0,1\}^n \to \{0,1\}^{\ell(k,n)}$ be a function, and let $F : 1^k \times \{0,1\}^n \times \{0,1\}^{m(k,n)} \to \{0,1\}^{s(k,n)}$ be a randomized function whose third argument is its random tape. We say that $G(1^k,x)$ is encoded by $F(1^k,x;r)$ if there exist a recovery algorithm $\mathsf{Rec} : 1^k \times \{0,1\}^{s(k,n)} \to \{0,1\}^{\ell(k,n)}$ and a randomized simulator algorithm $\mathsf{Sim} : 1^k \times \{0,1\}^{\ell(k,n)} \to \{0,1\}^{s(k,n)}$ that satisfy the following:*

- **Perfect correctness.** *For every k,n and $x \in \{0,1\}^n$, the error probabilities*

$$\Pr_r[\mathsf{Rec}(1^k, F(1^k,x;r)) \neq G(1^k,x)] \text{ and } \Pr[\mathsf{Rec}(1^k, \mathsf{Sim}(1^k, G(1^k,x))) \neq G(1^k,x)]$$

are both zero.[9]

- **Computational privacy.** *For every adversary $\mathcal{A}$ of complexity* $\mathrm{poly}(k)$ *we have that*

$$\left|\Pr[\mathcal{A}^{F(1^k,\cdot;\mathsf{U})}(1^k)=1]-\Pr[\mathcal{A}^{\mathsf{Sim}(1^k,G(1^k,\cdot))}(1^k)=1]\right|<\mathrm{neg}(k),$$

  where the oracles are defined as follows: Given x the first oracle returns a sample from $F(1^k,x;r)$ where the randomness r is chosen uniformly at random, and the second oracle returns a sample from $\mathsf{Sim}(1^k,G(1^k,x))$.

- **Efficiency.** *Typically,* efficiency *requires that both F and G are computable in time k^c for some constant c. In our context, we relax this requirement and say that the encoding is* efficient *if F and G are computable in time $(k+\ell)^c$ where $\ell(|x|)$ is the output length of $G(1^k,x)$. This means that the output length s of $F(1^k,x;r)$ is polynomial in $k+\ell(|x|)$. In addition, the complexity of* Rec *and* Sim *is assumed to be polynomial in the length of their inputs, and therefore it is also polynomial in $k+\ell(|x|)$.*

Let $\mathcal{G}(1^k,z,\vec{\mathsf{sk}})$ and $\mathcal{F}(1^k,(z,r),\vec{\mathsf{sk}})$ be a pair of KDM function ensembles with the same arity $t=t(k)$. We say that $\mathcal{G}$ is *encoded* by $\mathcal{F}$ if the function $\mathcal{G}(1^k,(z,\vec{\mathsf{sk}}))$ is encoded by the function $\mathcal{F}'(1^k,(z,\vec{\mathsf{sk}});r):=\mathcal{F}(1^k,(z,r),\vec{\mathsf{sk}})$, where r is treated as a random input of $\mathcal{F}'$.

2.4 Reductions among KDM-ensembles

We say that a KDM function ensemble $\mathcal{G}$ KDM-reduces to another KDM function ensemble $\mathcal{F}$ (in symbols $\mathcal{G}\leq_{\mathsf{KDM}}\mathcal{F}$) if there exists a transformation which converts an encryption scheme $\mathcal{E}$ that is $\mathcal{F}$-KDM secure to an encryption scheme $\widehat{\mathcal{E}}$ which is $\mathcal{G}$-KDM secure. Formally, such a (black-box) reduction is composed of (1) (construction) an encryption scheme $\widehat{\mathcal{E}}$ which is given an oracle access to the scheme $\mathcal{E}$; and (2) (security reduction) an efficient algorithm $\mathcal{B}$ such that for any $\mathcal{F}$-adversary $\mathcal{A}$ which attacks $\mathcal{E}$ with advantage α, the $\mathcal{G}$-adversary $\mathcal{B}^{\mathcal{A},\mathcal{E}}$ attacks the scheme $\widehat{\mathcal{E}}$ with polynomially related advantage (e.g., $\alpha/\mathrm{poly}(k)$). This definition can be instantiated with respect to all four different types. We say that the reduction is *type-preserving* if $\mathcal{B}^{\mathcal{A},\mathcal{E}}$ is always of the same type as $\mathcal{A}$ (i.e., $\mathcal{B}$ always ask the same type of queries that $\mathcal{A}$ asks in the KDM game.) Type preserving reduction extends KDM-security while being insensitive to the concrete setting which is being used. Formally,

Lemma 2.6 (KDM-reductions)**.** *Suppose that the KDM function ensemble $\mathcal{G}$ KDM-reduces to the ensemble $\mathcal{F}$ via a type-preserving reduction $(\widehat{\mathcal{E}},\mathcal{B})$. For every $T\in\{\mathsf{pub},\mathsf{sym}\}\times\{\mathsf{CCA},\mathsf{CPA}\}$, if the encryption scheme $\mathcal{E}$ is $(T,\mathcal{F})$-KDM secure then the scheme $\widehat{\mathcal{E}}^{\mathcal{E}}$ is $(T,\mathcal{G})$-KDM secure.*

3 KDM Reductions via Randomized Encoding

Our main theorem shows that randomized encoding gives rise to KDM reductions.

Theorem 3.1 (main theorem)**.** *Suppose that the KDM function ensemble $\mathcal{F}$ encodes the KDM function ensemble $\mathcal{G}$. Then, $\mathcal{G}$ KDM-reduces to $\mathcal{F}$ via a type-preserving reduction.*

[9] Previous definitions require only that the first quantity is zero, however, all known constructions (of perfectly-correct randomized encoding) satisfy the current (stronger) definition.

To prove the theorem we need to describe a construction and a security reduction. From now on, let Sim and Rec be the simulator and recovery algorithm which establish the encoding of $\mathcal{G}$ by $\mathcal{F}$.

Construction 3.2. *Given an oracle access to the encryption scheme $\mathcal{E} = (\mathsf{KG}, \mathsf{E}, \mathsf{D})$, we define the scheme $\widehat{\mathcal{E}}$ as follows*

$$\widehat{\mathsf{KG}}(1^k) = \mathsf{KG}(1^k) \qquad \widehat{\mathsf{E}}_{\mathsf{pk}}(M) = \mathsf{E}_{\mathsf{pk}}(\mathsf{Sim}(M)) \qquad \widehat{\mathsf{D}}_{\mathsf{sk}}(C) = \mathsf{Rec}(\mathsf{D}_{\mathsf{sk}}(C)),$$

where all algorithms (i.e., encryption, decryption, simulator and recovery) get the security parameter 1^k as an additional input.

It is not hard to show that $\widehat{\mathcal{E}}$ satisfies the syntactic requirements of encryption schemes. Indeed, the complexity of $\widehat{\mathsf{KG}}(1^k)$ is polynomial in the security parameter k, while the efficiency of the encoding ensures that the complexity of $\widehat{\mathsf{E}}_{\mathsf{pk}}(M)$ (resp., $\widehat{\mathsf{D}}_{\mathsf{sk}}(C)$) is polynomial in k and $|M|$ (resp., k and $|C|$). Correctness also follows easily as shown in the following lemma.

Lemma 3.3 (correctness)**.** *The decryption error of the scheme $\widehat{\mathcal{E}}$ is the same as the decryption error δ of $\mathcal{E}$, and so it is negligible.*

Proof. The probability that a message M is incorrectly decrypted is bounded by

$$\Pr_{(\mathsf{pk},\mathsf{sk})\xleftarrow{R}\mathsf{KG}(1^k), M'\xleftarrow{R}\mathsf{Sim}(M)}[\mathsf{D}_{\mathsf{sk}}(\mathsf{E}_{\mathsf{pk}}(M')) \neq M'] + \Pr[\mathsf{Rec}(M') \neq M],$$

since the second term is 0, due to the (perfect) correctness of the encoding, we can bound the above by $\max_{M'} \Pr[\mathsf{D}_{\mathsf{sk}}(\mathsf{E}_{\mathsf{pk}}(M')) \neq M'] \leq \delta(k)$, where M' ranges over the support of $\mathsf{Sim}(M)$ and $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{R} \mathsf{KG}(1^k)$. □

Next, we show that the security of $\widehat{\mathcal{E}}$ can be based on that of $\mathcal{E}$. Given an oracle access to a $(T, \mathcal{G})$ adversary $\mathcal{A}$ that attacks $\widehat{\mathcal{E}}$, we define a $(T, \mathcal{F})$ adversary $\mathcal{B}$ that attacks $\mathcal{E}$ by randomly choosing one of two strategies $\mathcal{B}^0$ and $\mathcal{B}^1$.

Reduction 3.4 (The adversary $\mathcal{B}^{\mathcal{A},\mathcal{E}}$)**.** *Toss a coin $\sigma \xleftarrow{R} \{0,1\}$. If $\sigma = 1$ invoke the following adversary $\mathcal{B}^1$:*

- ***Initialization:*** *$\mathcal{B}^1$ invokes $\mathcal{A}$. If $\mathcal{A}$ asks for the encryption keys then $\mathcal{B}^1$ makes a similar query and passes the answer to $\mathcal{A}$.*
- ***Encryption query:*** *If $\mathcal{A}$ makes an encryption query (i, M), for $i \in [t]$ and $M \in \{0,1\}^*$, then $\mathcal{B}^1$ samples $M' = \mathsf{Sim}(M)$, sends (i, M') as an encryption query (wrt to $\mathcal{E}$) and passes the answer of the challenger to $\mathcal{A}$.*
- ***KDM query:*** *If $\mathcal{A}$ makes a KDM query (i, z), for $i \in [t]$ and $z \in \{0,1\}^*$ (i.e., for the function $\mathcal{G}(1^k, z, \cdot)$), then the adversary $\mathcal{B}^1$ uniformly chooses randomness r and asks the KDM query $(i, (z, r))$ (i.e., for the randomized encoding $\mathcal{F}'(1^k, z, \cdot; r) = \mathcal{F}(1^k, (z, r), \cdot)$). The answer of the challenger is being sent to $\mathcal{A}$.*

- ***Decryption query:*** *If $\mathcal{A}$ makes a decryption query (i, C), then $\mathcal{B}^1$ checks that it is legal (by inspecting all previous encryption/KDM queries), and if so, (1) passes the same decryption query to the challenger, (2) applies the recovery algorithm* Rec *to the result, and (3) sends it back to $\mathcal{A}^1$.*

- ***Termination:*** *$\mathcal{B}^1$ terminates with the same output of $\mathcal{A}$.*

If $\sigma = 0$ then invoke the adversary $\mathcal{B}^0$. This adversary is similar to $\mathcal{B}^1$ except that encryption and KDM queries of $\mathcal{A}$ are both translated into encryption queries as follows: given an encryption query of $\mathcal{A}$ of the form (i, M) (resp., KDM query of the form (i, z)), the adversary $\mathcal{B}^0$ samples $M' = \mathsf{Sim}(0^\ell)$ and asks for the ciphertext $\mathsf{E}_{\mathsf{pk}_i}(M')$, where ℓ is the length of M (resp., output length of $\mathcal{G}(1^k, z, \cdot)$).[10] *At the end, $\mathcal{B}^0$ flips the output of $\mathcal{A}$ and terminates.*

Note that the above reduction is indeed type-preserving. Before we analyze the reduction, we need some notation. Let $V_{\mathcal{A},0}(k)$ (resp., $V_{\mathcal{A},1}(k)$) be the random variable which describes the view of $\mathcal{A}$ in the $\mathcal{G}$-KDM game with respect to $\widehat{\mathcal{E}}$ conditioned on the event that the challenger sets the challenge bit b to 0 (resp., 1). Similarly, for $b \in \{0,1\}$ and $\sigma \in \{0,1\}$ let $V_{\mathcal{A},\mathcal{B}^\sigma,b}(k)$ be the view of $\mathcal{A}$ as emulated by $\mathcal{B}^\sigma$ conditioned on the event that the challenge bit (in the $\mathcal{F}$-KDM game that $\mathcal{B}^\sigma$ plays) is b.

Let us first focus on the adversary $\mathcal{B}^1$. If the challenge bit b is 1 (i.e., when the challenger is in the "real-mode"), then the difference between the emulated view $V_{\mathcal{A},\mathcal{B}^1,1}(k)$ and the view of $\mathcal{A}$ in the actual KDM game $V_{\mathcal{A},1}(k)$, is only due to the difference in the way KDM queries are answered. In the real game answers to KDM queries are computed properly as $\widehat{\mathsf{E}}_{\mathsf{pk}_i}(\mathcal{G}(1^k, z, \vec{\mathsf{sk}})) = \mathsf{E}_{\mathsf{pk}_i}(\mathsf{Sim}(\mathcal{G}(1^k, z, \vec{\mathsf{sk}})))$, whereas in the emulated game they are computed by $\mathsf{E}_{\mathsf{pk}_i}(\mathcal{F}(1^k, (z, \mathsf{U}), \vec{\mathsf{sk}}))$. However, this difference should not be noticeable due to the privacy of the randomized encoding. Formally, let $\alpha_b(k)$ (resp., $\beta_b^\sigma(k)$) denote the probability that $\mathcal{A}$ (resp., $\mathcal{B}^\sigma$) guesses the challenge bit when it takes the value b. Then,

Lemma 3.5. $\left|\beta_1^1(k) - \alpha_1(k)\right| \leq \mathrm{neg}(k)$.

Proof. We define the following distinguisher $\mathcal{D}$ which, given an oracle access to either $\mathsf{Sim}(\mathcal{G}(1^k, \cdot, \cdot))$ or to $\mathcal{F}'(1^k, (\cdot, \cdot); \mathsf{U}) = \mathcal{F}(1^k, (\cdot; \mathsf{U}), \cdot)$, attempts to distinguish between the two. The adversary $\mathcal{D}$ emulates the challenger with challenge bit $b = 1$. It generates a key vector $(\mathsf{sk}_i, \mathsf{pk}_i)_{i \in [t]}$ by executing the key-generation algorithm $\mathsf{KG}(1^k)$ for t times. Then $\mathcal{D}$ invokes $\mathcal{A}$. If $\mathcal{A}$ asks a KDM query (i, z) then $\mathcal{D}$ calls its oracle with the value $(z, \vec{\mathsf{sk}})$. Let M denote the answer of the oracle. The distinguisher computes the ciphertext $C = \mathsf{E}_{\mathsf{pk}_i}(M)$ and sends the ciphertext C to $\mathcal{A}$. If $\mathcal{A}$ asks other types of queries such as public-key queries, encryption queries, and decryption queries, the distinguisher $\mathcal{D}$ answers them properly exactly as the real challenger does when it's in the real mode $b = 1$. (For the case of a decryption query (i, C), the distinguisher checks that it is legal by inspecting all previous KDM/encryption queries, and if so, sends $\mathsf{D}_{\mathsf{sk}_i}(C)$.) The distinguisher halts with output 1 if and only if $\mathcal{A}$ outputs 1.

Note that: (1) If $\mathcal{D}$ gets an oracle access to $\mathsf{Sim}(\mathcal{G}(1^k, \cdot, \cdot))$ then the view of $\mathcal{A}$ is distributed exactly as $V_{\mathcal{A},1}(k)$ and so in this case $\mathcal{D}$ outputs 1 with probability $\alpha_1(k)$; (2) If $\mathcal{D}$ gets an oracle access to $\mathcal{F}'(1^k, (\cdot, \cdot); \mathsf{U})$ then the view of $\mathcal{A}$ is distributed exactly as in $V_{\mathcal{A},\mathcal{B}^1,1}(k)$, and so in this case $\mathcal{D}$ outputs 1 with probability $\beta_1^1(k)$. Hence, by the privacy of the encoding, it follows that $\left|\beta_1^1(k) - \alpha_1(k)\right| \leq \mathrm{neg}(k)$. □

[10] Recall that the output length can be efficiently computed given z.

We would like to argue now that a similar thing happens in the "fake" mode when $b = 0$; namely, that $V_{\mathcal{A},\mathcal{B}^1,0}(k)$ is indistinguishable from $V_{\mathcal{A},0}(k)$, and therefore β_0^1 is close to α_0. However, when $\mathcal{A}$ attacks the original scheme (and the challenger is in the "fake" mode) his KDM queries are answered with $\widehat{\mathsf{E}}_{\mathsf{pk}_i}(0^\ell) = \mathsf{E}_{\mathsf{pk}_i}(\mathsf{Sim}(0^\ell))$, whereas in the game emulated by $\mathcal{B}^1$ these queries are answered by $\mathsf{E}_{\mathsf{pk}_i}(0^s)$, where $\ell = |\mathcal{G}(1^k, z, \vec{\mathsf{sk}})|$ and $s = |\mathcal{F}(1^k, (z; \mathsf{U}), \vec{\mathsf{sk}})|$. Although the privacy of the encoding ensures that the plaintexts are of the same length, i.e., $s = |\mathsf{Sim}(0^\ell)|$, the actual distributions of the plaintexts may differ, and so it may be the case that the two views are distinguishable. Intuitively, such a scenario would violate the ciphertext-indistinguishability of the encryption. To make this intuition formal, we need the adversary $\mathcal{B}^0$ which breaks the standard ciphertext-indistinguishability security of $\mathcal{E}$ whenever such a gap exists. As a result we will show that the average success probability of $\mathcal{B}^1$ and $\mathcal{B}^0$ is roughly half the success probability of $\mathcal{A}$. To this aim we prove the following

Lemma 3.6. *$\beta_1^0(k) = \alpha_0(k)$ and $\beta_0^0(k) + \beta_0^1(k) = 1$.*

Proof. First, we note that $V_{\mathcal{A},\mathcal{B}^0,1}(k)$, the view of $\mathcal{A}$ as emulated by $\mathcal{B}^0$ when the challenge bit $b = 1$, is identical to $V_{\mathcal{A},0}(k)$ the view of $\mathcal{A}$ in the real game when the challenge $b = 0$. Indeed, in both cases a KDM query (i, z) (resp., an encryption query (i, M)) is answered with $\widehat{\mathsf{E}}_{\mathsf{pk}_i}(0^\ell) = \mathsf{E}_{\mathsf{pk}_i}(\mathsf{Sim}(0^\ell))$ where ℓ is the output length of $\mathcal{G}(1^k, z, \cdot)$ (resp., $\ell = |M|$). Since $\mathcal{B}^0$ flips the output of $\mathcal{A}$ it follows that $\beta_1^0(k)$ equals to $\alpha_0(k)$.

To prove the second equality we first claim that $V_{\mathcal{A},\mathcal{B}^0,0}(k)$, the view of $\mathcal{A}$ when emulated by $\mathcal{B}^0$ when the challenge bit $b = 0$, is identically distributed to $V_{\mathcal{A},\mathcal{B}^1,0}(k)$, the view of $\mathcal{A}$ as emulated by $\mathcal{B}^1$ when the challenge bit $b = 0$. Indeed, the only difference is that in the first case KDM queries (i, z) are answered by $\mathsf{E}(0^{|\mathsf{Sim}(\mathcal{G}(1^k, z, \vec{\mathsf{sk}}))|})$, while in the second case the answer is $\mathsf{E}(0^{|\mathcal{F}(1^k, (z;r), \mathsf{sk})|})$. Since z and k are fixed, the output lengths of $\mathcal{F}(1^k, z; (r, \cdot))$ and $\mathsf{Sim}(\mathcal{G}(1^k, z, \cdot))$ are fixed and equal, and so the claim follows. The claim implies that $\beta_0^0(k) + \beta_0^1(k) = 1$, as $\mathcal{B}^1$ outputs the outcome of $\mathcal{A}$, and $\mathcal{B}^0$ flips it. □

By combining the two lemmas (3.5 and 3.6), it follows that the advantage $\beta = (\beta_1^1 + \beta_0^1 + \beta_0^0 + \beta_1^0)/4 - \frac{1}{2}$ of $\mathcal{B}$ is at least $\frac{1}{2}\alpha - \mathrm{neg}(k)$ where $\alpha = \frac{1}{2}(\alpha_1 + \alpha_0) - \frac{1}{2}$ is the advantage of $\mathcal{A}$. Hence, we established the correctness of the reduction.

Theorem 3.7. *If $\mathcal{A}$ is an efficient adversary that breaks $\widehat{\mathcal{E}}$ with respect to $\mathcal{G}$ with advantage $\alpha(k)$, then the adversary $\mathcal{B}^{\mathcal{A},\mathcal{E}}$ breaks $\mathcal{E}$ with respect to $\mathcal{F}$ with advantage $\beta(k) \geq \alpha(k)/2 - \mathrm{neg}(k)$.*

Remark 3.8. *By inspecting the above proof, we can see that Theorem 3.1 tolerates the following relaxations:*

1. *Assume that $\mathcal{G}(1^k, (z, \vec{\mathsf{sk}}))$ is encoded by the function $\mathcal{F}'(1^k, (z, \vec{\mathsf{sk}}); r)$ and that the ensemble $\mathcal{F}(1^k, z', \vec{\mathsf{sk}})$ is indexed by z'. Then, the reduction works as long as there exists an efficiently computable translation function $\rho : (1^k, z, r) \mapsto z'$ such that $\mathcal{F}(1^k, \rho(1^k, z, r), \vec{\mathsf{sk}}) = \mathcal{F}'(1^k, (z, \vec{\mathsf{sk}}); r)$. (Recall that the original definition of encoding in Section 2.3 corresponds to the special case where ρ is the identity function.)*

2. *The proof goes through even if the encoding itself makes use of the underlying encryption scheme $\mathcal{E}$ as long as this usage is done in a fully black-box way (the same holds for any cryptographic primitive which can be based on $\mathcal{E}$ via a black-box reduction e.g., one-way function).*

More precisely, Theorem 3.1 holds (i.e., lead to black-box KDM reduction/construction) as long as the security of the encoding reduces to the security of the underlying primitive (i.e., $\mathcal{E}$) via a black-box reduction, and as long as the simulator, decoder, and the translation function ρ can be implemented given a black-box access to the underlying primitive.

3. *The reduction is insensitive to the number of keys in the system. Specifically, under the refined notion of $\mathsf{KDM}^{(\tau)}$ security (Remark 2.2), the proof essentially shows that if $\mathcal{F}$ encodes the KDM function ensemble $\mathcal{G}$, then, for every τ, $\mathcal{G}$ $\mathsf{KDM}^{(\tau)}$-reduces to $\mathcal{F}$ via a type-preserving reduction.*

4 Completeness of Projections

In [6] it is shown that Yao's garbled circuit technique allows to encode any efficiently computable function by a decomposable encoding in which every bit depends on at most a single bit of the deterministic input. This means that, for every fixed randomness, the encoding is a projection. (See Section 2.2 for a definition.) Formally,

Fact 4.1 ([6])**.** *Let $\varepsilon > 0$ be an arbitrarily small constant. Every function $G(1^k, x)$ of circuit-size $a(k)$ can be encoded by a function $F(1^k, x; r)$ with the following properties:*

1. *The simulator and decoder use a black-box access to a symmetric encryption (equivalently, to a one-way function).*
2. *For every fixed randomness r, the resulting function $F_{k,r}(x) = F(1^k, x; r)$ is a* projection *function of output length $a(k)^{1+\varepsilon}$.*
3. *The mapping from the circuit of $G(1^k, \cdot)$ to the circuit of $F_{k,r}$ is efficiently computable given a black-box access to the symmetric encryption scheme.*
4. *The security of the encoding reduces to the security of the symmetric encryption scheme via a black-box reduction.*

By combining this fact with Theorem 3.1 we get the following:

Proposition 4.2 (Completeness of projections)**.** *Let $\mathcal{G}_k^t$ be a t-ary KDM ensemble with output length $\ell(|z|)$ where $t(\cdot), \ell(\cdot)$ are polynomials.*

- *If $\mathcal{G}_k^t$ is efficiently computable in time k^c then $\mathcal{G}_k^t \leq_{\mathsf{KDM}} \Pi_{k,k^{c+\varepsilon}}^t$, where $\varepsilon > 0$ is an arbitrarily small constant and $\Pi_{k,q}^t$ is the t-ary ensemble of projections of output length q.*
- *If $\mathcal{G}_k^t$ is output efficiently computable then $\mathcal{G}_k^t \leq_{\mathsf{KDM}} \Pi_k^t$ where $\Pi_k^t = \bigcup_{a \in \mathbb{N}} \Pi_{k,k^a}^t$.*

Moreover, the reductions are type preserving.

Hence, one can upgrade KDM security from (almost) the weakest KDM function ensemble to the very powerful notion of p-length-dependent KDM security.

Proof. Fix some $\varepsilon > 0$ and let $\mathcal{G}(1^k, z, \vec{\mathsf{sk}})$ be the evaluation algorithm of $\mathcal{G}^t_k$ whose complexity is k^c. By applying Fact 4.1, we obtain an encoding $\mathcal{F}'(1^k, (z, \vec{\mathsf{sk}}); r)$ such that for every fixing of r the resulting function is in $\Pi^t_{k,k^{(1+\varepsilon)c}}$. By applying Theorem 3.1 (together with Remark 3.8), it follows that $\mathcal{G}^t_k \leq_{\mathsf{KDM}} \Pi^t_{k,k^{(1+\varepsilon)c}}$. The same argument holds if $\mathcal{G}^t_k$ is output efficiently computable, except that now $\mathcal{G}^t_k$ is computable in time $(k + \ell(|z|))^c$ and so the encoding (with some fixed randomness) is a projection of output length $(k + \ell(|z|))^{c(1+\varepsilon)}$ which falls into Π^t_k. □

Specifically, (p, ℓ)-bounded circuit-size KDM security reduces to KDM-security with respect to polynomially bounded projections (of output length $q(k) = p^{1+\varepsilon}(k)$) and p-length-dependent KDM security reduces to KDM-security with respect to projections of arbitrary output length.

In the case of CPA KDM security, one can actually derive KDM-security with respect to projections of arbitrary output length (i.e., Π^t_k) from single-output projections $\Pi^t_{k,1}$.

Lemma 4.3 (Completeness of single-output projections for CPA-KDM)**.** *For every polynomial $t(\cdot)$, we have $\Pi^t_k \leq_{\mathsf{KDM}} \Pi^t_{k,1}$, where the reduction holds for both $(\mathsf{sym}, \mathsf{CPA})$ and $(\mathsf{pub}, \mathsf{CPA})$ types.*

Proof. The proof follows by simple concatenation: the new encryption/decryption algorithms encrypts/decrypts the message/ciphertext by applying the original encryption/decryption algorithm in a bit by bit manner. Hence, a KDM query in Π^t_{k,k^a} for the new scheme can be emulated by k^a KDM queries in $\Pi^t_{k,1}$ for the original scheme. □

As shown in [10], we can use the standard encrypt-then-MAC transformation to upgrade the security of a scheme that satisfies $(\mathsf{sym}, \mathsf{CPA})$-KDM security into a scheme that satisfies $(\mathsf{sym}, \mathsf{CCA})$-security with respect to the same KDM class. A similar result was proven for the public-key setting by [17] via the Naor-Yung double-encryption paradigm (which relies on the existence of NIZK). Hence, by Proposition 4.2 and Lemma 4.3, we have:

Corollary 4.4 (KDM Collapse)**.** *For every polynomials t and p, there exists a $\Pi^t_{k,1}$-KDM secure scheme if and only if there exists a t-ary p-length-dependent KDM secure encryption scheme. This holds unconditionally for the KDM types $(\mathsf{sym}, \mathsf{CPA})$, $(\mathsf{sym}, \mathsf{CCA})$, and $(\mathsf{pub}, \mathsf{CPA})$, and it holds for $(\mathsf{pub}, \mathsf{CCA})$ assuming the existence of non-interactive zero-knowledge proof system for* **NP**.

We remark that all the known constructions of affine-KDM secure encryption schemes [13, 5, 14] can be adapted to yield KDM security with respect to single-output projections (see Appendix A). Hence, we get p-length-dependent $(\mathsf{pub}, \mathsf{CPA})$-KDM (resp., $(\mathsf{sym}, \mathsf{CCA})$) based on the DDH, LWE, QR, or DCR assumptions (resp., LPN assumption), which can be boosted into $(\mathsf{pub}, \mathsf{CCA})$-KDM assuming the existence of NIZK for **NP**. Furthermore, the schemes based on [13, 5] remain secure even when there is an arbitrary number of keys in the system. (See Remarks 2.2 and 3.8).

5 On Full KDM Security

In this section, we study the possibility of constructing a scheme which satisfies KDM security for the class of all efficiently computable functions. In [11] it was shown that such a scheme can be constructed based on the existence of cyclic-secure fully homomorphic encryption (FHE) [23]. We show that a similar assumption is inherently required for full KDM security which is also *simulatable*. For simplicity, we focus on the case of arity $t = 1$ and single-query adversaries.

A public-key encryption scheme $\mathcal{E} = (\mathsf{KG}, \mathsf{E}, \mathsf{D})$ is simulatable $\mathcal{F}$-KDM secure if there exists a polynomial-time simulator S such that for every $(\mathsf{sk}, \mathsf{pk}) \in \mathsf{KG}(1^k)$, and every circuit family $f_k \in \mathcal{F}_k$ of size $\mathrm{poly}(k)$, the ensemble $S(\mathsf{pk}, f_k)$ is indistinguishable from $\mathsf{E}_{\mathsf{pk}}(f_k(\mathsf{sk}))$. (Note that this means that the distinguisher holds the secret-key sk.) The notions of *simulatable circular-security* and *simulatable full-KDM security* correspond to the two extreme cases where $\mathcal{F}$ contains only the identity function, and $\mathcal{F}$ contains all functions.

An FHE allows to translate encryptions of a message M into an encryption of a related message $h(M)$ for any polynomial-size circuit h. More formally, we say that $\mathcal{E}$ is *fully homomorphic* if there exists an efficient algorithm Eval such that for every $(\mathsf{sk}, \mathsf{pk}) \in \mathsf{KG}(1^k)$, every circuit family $\{h_k\}$ of size $\mathrm{poly}(k)$, and every sequence of messages $M_k \in \{0,1\}^{\mathrm{poly}(k)}$, the ensemble $\mathsf{Eval}(\mathsf{pk}, h_k, \mathsf{E}_{\mathsf{pk}}(M_k))$ is computationally indistinguishable from the ensemble $\mathsf{E}_{\mathsf{pk}}(h_k(M_k))$.

In [11], it was shown that if an encryption scheme is both simulatable circular-secure and fully-homomorphic then it is also simulatable fully-KDM secure. We show that the other direction holds as well, and so the two notions are equivalent.

Proposition 5.1. *Any simulatable fully-KDM secure encryption scheme is also fully-homomorphic circular-secure.*

Proof. Given a simulatable fully-KDM secure encryption scheme $(\mathsf{KG}, \mathsf{E}, \mathsf{D})$ with simulator S, we define $\mathsf{Eval}(\mathsf{pk}, h, C)$ by invoking S on the pair $(\mathsf{pk}, f_{h,C})$ where $f_{h,C}$ is the mapping $\mathsf{sk} \mapsto h(\mathsf{D}_{\mathsf{sk}}(C))$. Note that the circuit size of $f_{h,C}$ is polynomial in the circuit size of h (since D is efficient). Also, by definition, we have for every $(\mathsf{sk}, \mathsf{pk}) \in \mathsf{KG}(1^k)$, sequence $\{M_k\}$ and sequence $\{h_k\}$,

$$\begin{aligned} \mathsf{Eval}(\mathsf{pk}, h_k, \mathsf{E}_{\mathsf{pk}}(M_k)) &\equiv S(\mathsf{pk}, f_{h_k, \mathsf{E}_{\mathsf{pk}}(M_k)}) \\ &\stackrel{c}{\equiv} \mathsf{E}_{\mathsf{pk}}(h_k(\mathsf{D}_{\mathsf{sk}}(\mathsf{E}_{\mathsf{pk}}(M_k)))) \\ &\equiv \mathsf{E}_{\mathsf{pk}}(h_k(M_k)), \end{aligned}$$

where $\equiv$ (resp., $\stackrel{c}{\equiv}$) denotes statistical (resp., computational) indistinguishability. □

Let us waive the simultability requirement, and move back to the standard notion of KDM security. We show that if an encryption scheme $\mathcal{E} = (\mathsf{KG}, \mathsf{E}, \mathsf{D})$ provides KDM security gainst a function which is slightly "stronger" than its decryption algorithm D, then $\mathcal{E}$ is actually fully-KDM secure. This is done by observing that Gentry's "bootstrapping technique" can be adapted to the KDM setting.

Proposition 5.2. *Let $T \in \{(\mathsf{pub}, \mathsf{CPA}), (\mathsf{sym}, \mathsf{CPA})\}$, and let $\mathcal{E} = (\mathsf{KG}, \mathsf{E}, \mathsf{D})$ be T-KDM secure encryption with respect to single-output projections and with respect to the function family $\mathcal{F}_k = \{f_{C_1,C_2} : \mathsf{sk} \mapsto \mathrm{NAND}(\mathsf{D}_{\mathsf{sk}}(C_1), \mathsf{D}_{\mathsf{sk}}(C_2))\}$, where C_1, C_2 range over $\{0,1\}^{p(k)}$ and $p(k)$ is the length of an encryption of one-bit message under secret-key of length k. Then, $\mathcal{E}$ is fully KDM secure of type T.*

Proof. In the CPA setting it suffices to prove full KDM security with respect to all circuits of single output. We show how to convert an attacker which sends arbitrary KDM queries into one which uses only queries from $\mathcal{F}_k$. Let h be a circuit of size s, which is, without loss of generality, composed of NAND gates, and let h_i denote the function computed by the i-th gate of h, where gates are ordered under some topological ordering. We translate a KDM query for h into s KDM calls to $\mathcal{F}_k$ by traversing the circuit from bottom to top in a gate by gate manner preserving the following

invariant: The i-th query will be answered by a ciphertext C_i such that, if the oracle is in the real mode $C_i = \mathsf{E}_{\mathsf{pk}}(h_i(\mathsf{sk}))$ and if it is in the fake mode $C_i = \mathsf{E}_{\mathsf{pk}}(0)$. For an input gate, this can be achieved directly by making a single KDM query with a single-output projection. To do this for an internal gate h_ℓ whose input wires are connected to h_i and h_j for some $i, j < \ell$, we use a KDM query to f_{C_i,C_j}. It is not hard to see that the invariant holds, and therefore the claim follows. □

Acknowledgement. We thank Iftach Haitner, Yuval Ishai, and the anonymous referees for their helpful comments.

References

[1] Abadi, M., Rogaway, P.: Reconciling two views of cryptography (the computational soundness of formal encryption). Journal of Cryptology 20(3), 395 (Jul 2007)

[2] Acar, T., Belenkiy, M., Bellare, M., Cash, D.: Cryptographic agility and its relation to circular encryption. In: Advances in Cryptology – EUROCRYPT 2010. pp. 403–422 (2010)

[3] Adão, P., Bana, G., Herzog, J., Scedrov, A.: Soundness and completeness of formal encryption: The cases of key cycles and partial information leakage. Journal of Computer Security 17(5), 737–797 (2009)

[4] Applebaum, B.: Key-dependent message security: Generic amplification and completeness theorems. Cryptology ePrint Archive, Report 2010/513 (2010)

[5] Applebaum, B., Cash, D., Peikert, C., Sahai, A.: Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In: Advances in Cryptology – CRYPTO 2009. pp. 595–618 (2009)

[6] Applebaum, B., Ishai, Y., Kushilevitz, E.: Computationally private randomizing polynomials and their applications. Journal of Computional Complexity 15(2), 115–162 (2006)

[7] Applebaum, B., Ishai, Y., Kushilevitz, E.: Cryptography in NC^0. SIAM Journal on Computing 36(4), 845–888 (2006)

[8] Backes, M., Dürmuth, M., Unruh, D.: OAEP is secure under key-dependent messages. In: Advances in Cryptology – ASIACRYPT 2008. pp. 506–523 (2008)

[9] Amos Beimel and Anna Gál. On arithmetic branching programs. *Journal of Computer and System Sciences*, 59(2):195–220, 1999.

[10] Backes, M., Pfitzmann, B., Scedrov, A.: Key-dependent message security under active attacks - BRSIM/UC-soundness of symbolic encryption with key cycles. In: Proceedings of 20th IEEE Computer Security Foundation Symposium (CSF)(2007)

[11] Barak, B., Haitner, I., Hofheinz, D., Ishai, Y.: Bounded key-dependent message security. In: Advances in Cryptology – EUROCRYPT 2010. pp. 423–444 (2010)

[12] Black, J., Rogaway, P., Shrimpton, T.: Encryption-scheme security in the presence of key-dependent messages. In: SAC 2002: 9th Annual International Workshop on Selected Areas in Cryptography. pp. 62–75 (2002)

[13] Boneh, D., Halevi, S., Hamburg, M., Ostrovsky, R.: Circular-secure encryption from decision Diffie-Hellman. In: Advances in Cryptology – CRYPTO 2008. pp. 108–125 (2008)

[14] Brakerski, Z., Goldwasser, S.: Circular and leakage resilient public-key encryption under subgroup indistinguishability (or: Quadratic residuosity strikes back). In: Advances in Cryptology – CRYPTO 2010. pp. 1–20 (2010)

[15] Brakerski, Z., Goldwasser, S., Kalai, Y.: Circular-secure encryption beyond affine functions. In: TCC 2011: 8th Theory of Cryptography Conference(2011)

[16] Brakerski, Z., Vaikuntanathan, V.: Efficient Fully Homomorphic Encryption from (Standard) LWE. In: 52nd Annual Symposium on Foundations of Computer Science (FOCS). pp. 97–106 (2011)

[17] Camenisch, J., Chandran, N., Shoup, V.: A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In: Advances in Cryptology – EUROCRYPT 2009. pp. 351–368 (2009)

[18] Ronald Cramer, Serge Fehr, Yuval Ishai, and Eyal Kushilevitz. Efficient multi-party computation over rings. In: Advances in Cryptology – EUROCRYPT 2003. pages 596–613, 2003.

[19] Camenisch, J., Lysyanskaya, A.: An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In: Advances in Cryptology – EUROCRYPT 2001. pp. 93–118 (2001)

[20] van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully homomorphic encryption over the integers. In: Advances in Cryptology – EUROCRYPT 2010. pp. 24–43 (2010)

[21] Dolev, D., Dwork, C., Naor, M.: Non-malleable cryptography (extended abstract). In: 23rd Annual ACM Symposium on Theory of Computing (STOC). pp. 542–552 (1991)

[22] Even, S., Goldreich, O., Lempel, A.: A randomized protocol for signing contracts. Communications of the Association for Computing Machinery 28 (1985)

[23] Gentry, C.: Fully homomorphic encryption using ideal lattices. In: 41st Annual ACM Symposium on Theory of Computing (STOC). pp. 169–178 (2009)

[24] Gentry, C., Halevi, S.: Fully Homomorphic Encryption without Squashing Using Depth-3 Arithmetic Circuits. In: 52nd Annual Symposium on Foundations of Computer Science (FOCS). pp. 107–109 (2011)

[25] Gertner, Y., Kannan, S., Malkin, T., Reingold, O., Viswanathan, M.: The relationship between public key encryption and oblivious transfer. In: 41st Annual Symposium on Foundations of Computer Science (FOCS)(2000)

[26] Goldreich, O.: Foundations of Cryptography: Basic Tools. Cambridge University Press (2001)

[27] Goldreich, O.: Foundations of Cryptography: Basic Applications. Cambridge University Press (2004)

[28] Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences 28(2), 270–299 (1984)

[29] Haitner, I., Holenstein, T.: On the (im)possibility of key dependent encryption. In: TCC 2009: 6th Theory of Cryptography Conference. pp. 202–219 (2009)

[30] Halevi, S., Krawczyk, H.: Security under key-dependent inputs. In: ACM CCS 07: 14th Conference on Computer and Communications Security. pp. 466–475 (2007)

[31] Hofheinz, D., Unruh, D.: Towards key-dependent message security in the standard model. In: Advances in Cryptology – EUROCRYPT 2008. pp. 108–126 (2008)

[32] Impagliazzo, R., Rudich, S.: Limits on the provable consequences of one-way permutations. In: Advances in Cryptology – CRYPTO'88. pp. 8–26 (1988)

[33] Ishai, Y., Kushilevitz, E.: Randomizing polynomials: A new representation with applications to round-efficient secure computation. In: 41st Annual Symposium on Foundations of Computer Science (FOCS). pp. 294–304 (2000)

[34] Ishai, Y., Kushilevitz, E.: Perfect constant-round secure computation via perfect randomizing polynomials. In: ICALP 2002: 29th International Colloquium on Automata, Languages and Programming. pp. 244–256 (2002)

[35] Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: 22nd Annual ACM Symposium on Theory of Computing (STOC). pp. 427–437 (1990)

[36] Rabin, M.: Digitalized signatures and public key functions as intractable as factoring. Tech. Rep. 212, LCS, MIT (1979)

[37] Rackoff, C., Simon, D.R.: Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In: Advances in Cryptology – CRYPTO'91. pp. 433–444 (1991)

[38] Vaikuntanathan, V.: Computing Blind folded: New Developments in Fully Homomorphic Encryption. In: 52nd Annual Symposium on Foundations of Computer Science (FOCS). pp. 5–16 (2011)

[39] Yao, A.C.: How to generate and exchange secrets. In: 27th Annual Symposium on Foundations of Computer Science (FOCS). pp. 162–167 (1986)

A From affine functions to projections

Converting affine-security to security under single-output projections is immediate if the affine functions are defined over the binary field $\mathbb{F}_2$ (as in the LPN based scheme of [17] or the QR based schemes of [14]), but can also be established in more general cases, which capture most known schemes, as follows.

Suppose that we have a scheme $\mathcal{E}$ which encrypts ring elements $M \in \mathcal{R}$, using a secret-key $\mathsf{sk} = (\mathsf{sk}_i)_{i\in[n]} \in \mathcal{R}^n$. The scheme provides KDM security with respect to the class of affine functions from $\mathcal{R}^n$ to $\mathcal{R}$ namely:

$$\mathcal{L} = \{L_{a,b} | a \in \mathcal{R}^n, b \in \mathcal{R}\}, \quad \text{where } L_{a,b} : \mathsf{sk} \mapsto b + \sum_i a_i \cdot \mathsf{sk}_i.$$

BHHO like schemes. Assume that each key element sk_i is either the additive identity element $\mathbf{0}$ of the ring $\mathcal{R}$ or the multiplicative identity element $\mathbf{1}$ of $\mathcal{R}$. Let us represent each key element sk_i by a single bit $\langle \mathsf{sk} \rangle_i$ in the natural way. In this case, the ensemble of bit-wise projections is a sub-class of affine functions over $\mathcal{R}$. Indeed, the projection $f_{i,\sigma}(\langle \mathsf{sk} \rangle) = \langle \mathsf{sk} \rangle_i \oplus \sigma$ can be written as sk_i if $\sigma = 0$, and as $\mathbf{1} - \mathsf{sk}_i$ if $\sigma = 1$. Hence, KDM security with respect to projections follows immediately from affine-KDM security. This case captures the DDH-based schemes of [13] and the schemes of [14] which are based on the DCR or QR assumptions (or more generally on the subgroup indistinguishability assumption).

Efficiently computable bit-wise representation. We proceed with a more general approach. Assume the the secret key $\mathsf{sk} \in \mathcal{R}^n$ is represented by a k-bit string denoted by $\langle \mathsf{sk} \rangle = (\langle \mathsf{sk} \rangle_1, \ldots, \langle \mathsf{sk} \rangle_k)$. Furthermore, assume that the mapping from sk to each bit of the representation $\langle \mathsf{sk} \rangle$ can be computed by a polynomial-size arithmetic formula (or, more generally, arithmetic branching program, see [9, 18]) over $\mathcal{R}$. Then, the mappings $f_{i,0} : \mathsf{sk} \mapsto \langle \mathsf{sk} \rangle_i$ and $f_{i,1} : \mathsf{sk} \mapsto 1 - \langle \mathsf{sk} \rangle_i$ can also be computed by a polynomial-size formula. Hence, by [18], there exists a perfect (universal) RE $\hat{f}_{i,\sigma}(\mathsf{sk}; r)$ such that for every fixed choice of r, $\hat{f}_{r,i,\sigma}(\mathsf{sk}) = \hat{f}_{i,\sigma}(\mathsf{sk}; r)$ is an affine function over $\mathcal{R}$. Therefore, by Theorem 3.1, the security of the scheme can be amplified to hold with respect to single-output projections.

This approach is useful, for example, when the ring is of polynomial size in the security parameter (as in the LWE-based scheme of [5]). In this case, one can trivially compute the (standard) binary-decomposition of ring elements by a polynomial size formula. For example, if the multiplicative order of the ring is p then the i-th bit of the representation of a ring element x can be computed by the formula $\prod_r (x - r)^{p-1}$ where r ranges over all polynomially-many elements in $\mathcal{R}$ having 0 in the i-th coordinate of their binary decomposition.

This example can be easily extended to the case where the ring $\mathcal{R}$ can be decomposed into several rings of polynomial-size. For example, consider the ring $\mathcal{R} = \mathbb{Z}_p$ where $p = \prod p_i$, and the p_i's are polynomially bounded co-primes. Then, by the CRT, we can first project an element $x \in \mathbb{Z}_p$ to the sub-ring $\mathbb{Z}_{p_1} \times 1 \times \ldots \times 1$ via the formula x^{q/p_1}, and then recover the bit-representation inside $\mathbb{Z}_{p_1}$ via the previous brute-force formula. By repeating the process for each factor p_i, and accumulating the sub-ring representations, we obtain a bitwise representation of x. A similar approach can be used by decomposing the multiplicative (resp., additive) group of the ring to polynomially bounded multiplicative (resp., additive) subgroups. This can be done, for example, if $\mathbb{Z}_q$ is an exponentially large ring whose multiplicative order $\phi(q)$ factors into polynomially bounded co-primes $p_1, \ldots, p_\ell$ (e.g., when q is a prime). More generally, it suffices to break the set $\mathcal{R}$ into a product of polynomially bounded sets $S_1 \times \ldots \times S_k$ (not necessarily sub-rings) such that the i-th coordinate of an element $x \in \mathcal{R}$ can be computed by arithmetic formula over $\mathcal{R}$. We believe that such strategy can be applied to the scheme of [16] (at least for some range of the parameters).