Circular-Secure Encryption Beyond Affine Functions

Zvika Brakerski* Shafi Goldwasser† Yael Kalai‡

October 1, 2009

Abstract

We show that for any constant $d \in \mathbb{N}$, there exists a public-key encryption scheme that can securely encrypt any function f of its own secret-key, assuming f can be expressed as a polynomial of total degree-d. Such a scheme is said to be key-dependent message (KDM) secure w.r.t. degree-d polynomials. We also show that there exists a public-key encryption scheme that is KDM secure w.r.t. all Turing machines of bounded description length and bounded running time. The security of such public-key schemes can be based either on the standard decision Diffie-Hellman (DDH) assumption or on the learning with errors (LWE) assumption (with certain parameters settings).

In the case of functions that can be expressed as degree-d polynomials, we show that the resulting schemes are also secure with respect to *key cycles*. Specifically, given a polynomial number n of key pairs, the schemes can securely encrypt a degree-d polynomial whose variables are the collection of coordinates of all n secret-keys.

Our key idea is a general transformation that amplifies KDM security. The transformation takes an encryption scheme that is KDM secure w.r.t. *some* functions even when the secret keys are weak (i.e. chosen from an arbitrary distribution with entropy k), and outputs a scheme that is KDM secure w.r.t. a *richer* class of functions. The resulting scheme may no longer be secure with weak keys. Thus, in some sense, this transformation converts security with weak keys into amplified KDM security.

*Weizmann Institute of Science, zvika.brakerski@weizmann.ac.il.

†Weizmann Institute of Science and Massachusetts Institute of Technology, shafi@theory.csail.mit.edu.

‡Microsoft Research, yael@microsoft.com.

1 Introduction

Secure encryption is one of the most fundamental tasks in cryptography, and significant work has gone into defining and attaining it. All commonly accepted definitions for secure encryption [GM84, RS91, BN00, BR00, DDN00, KY00, Kra01], assume that the plaintext messages to be encrypted are independent of the secret decryption keys. However, over the years, it was observed that in some situations the plaintext messages *do* depend on the secret keys. For example, a backup system may store the backup encryption key on disk and then encrypt the entire disk, including the key, and backup the result. Other examples arise in the context of anonymous credential systems, as observed by Camenisch and Lysyanskaya [CL01], and in the context of "axiomatic secuirty" [ABHS05, LC03].

Security in this more demanding setting was termed *key-dependent message security* (KDM security) by Black, Rogoway and Shrimpton in [BRS02], who defined KDM security both in the symmetric and in the public-key settings. In the public-key setting, the adversary is given public keys $\mathrm{pk}_1, \ldots, \mathrm{pk}_n$ and can access an oracle $\mathcal{O}$ that upon receiving a query (i, g), where g is a polynomial size circuit and $i \in [n]$ is an index, returns an encryption of $g(\mathrm{sk}_1, \ldots, \mathrm{sk}_n)$ under the public key pk_i. The scheme is $\mathrm{KDM}^{(n)}$ secure, where n is the number of public keys, if the adversary cannot distinguish between the oracle $\mathcal{O}$ and an oracle that always returns an encryption of (say) the all-zero string. In particular, in $\mathrm{KDM}^{(1)}$-security, the adversary is given a single public key pk and can ask for encryptions (under pk) of functions of the corresponding secret key sk.

A more restrictive way to define KDM security is w.r.t. a class of functions. An encryption scheme is said to be $\mathrm{KDM}^{(n)}$ *secure w.r.t. functions in* $\mathcal{F}$, if the oracle $\mathcal{O}$ only answers queries (i, f) for $f \in \mathcal{F}$ and $i \in [n]$.

In the last few years, the notion of KDM security has been extensively studied [HK07, BPS07, BDU08, HU08, BHHO08, HH09, CCS09, ACPS09]. Without resorting to random oracles, constructing an encryption scheme that is KDM-secure w.r.t. *all* efficient functions of the secret key (either in the symmetric or in the public-key setting) is a long-standing open problem. Significant progress was recently made by Boneh, Halevi, Hamburg and Ostrovsky [BHHO08], who constructed a public-key encryption scheme that is $\mathrm{KDM}^{(n)}$ secure w.r.t. all affine functions (more precisely "affine in the exponent"), under the DDH assumption, for any polynomial n. This fundamental result was followed by a work of Applebaum, Cash, Peikert and Sahai [ACPS09] who proved that a variation of Regev's scheme [Reg05] is also KDM secure w.r.t. all affine functions, under the LWE assumption. They also constructed a very efficient symmetric encryption scheme which is KDM secure w.r.t. all affine functions, under the learning parity with noise (LPN) assumption.

A natural question to ask is: do there exist encryption schemes (either in the symmetric or in the public-key setting) that are KDM secure w.r.t. a richer class of functions?

Heitner and Holenstein [HH09] gave impossibility results with regards to black-box construction of $\mathrm{KDM}^{(1)}$-secure encryption (even in the symmetric case). They showed that $\mathrm{KDM}^{(1)}$-security w.r.t. poly-wise independent functions is not black-box reducible to one-way trapdoor permutations, and also that $\mathrm{KDM}^{(1)}$-security w.r.t. *all* functions is not black-box reducible to essentially any cryptographic assumption.

1.1 Our Results

We provide a general transformation that amplifies KDM security. Throughout this work, we restrict our attention to public-key encryption schemes in which the key-generation algorithm works by first sampling a secret key and then applying some, possibly randomized, function to produce the public key. Many known encryption schemes have this property, e.g. [RSA78, Gam84, Reg05, BHHO08, ACPS09] and others. We say that an encryption scheme is *entropy-k* KDM-*secure* if it is KDM-secure even when the secret-key is sampled from an arbitrary distribution with min-entropy k, and the computation of the public-key is performed with perfect randomness.[1] Our transformation starts with an encryption scheme $\mathcal{E} = (G, E, D)$ that is entropy-k KDM$^{(n)}$ secure w.r.t. some class of functions $\mathcal{F}$, and converts it into another scheme $\mathcal{E}^* = (G^*, E^*, D^*)$, which is KDM$^{(n)}$ secure w.r.t. a larger class of functions $\mathcal{F}'$.

Theorem 1.1 (informal)**.** *Let $\mathcal{E} = (G, E, D)$ be a public-key encryption scheme that is entropy-k KDM$^{(n)}$-secure w.r.t. a function class $\mathcal{F}$. Let $\mathcal{S}$ denote the space of the secret keys of $\mathcal{E}$, and let $\mathcal{K}$ be any set of size at least 2^k. Then for every deterministic, efficiently computable and injective mapping $\alpha : \mathcal{K} \to \mathcal{S}$ there exists an encryption scheme $\mathcal{E}^*_\alpha = (G^*, E^*, D^*)$, whose secret key, sk^*, are chosen at random from $\mathcal{K}$, such that $\mathcal{E}^*_\alpha$ is KDM$^{(n)}$ secure w.r.t. the function class $\mathcal{F}' = \mathcal{F} \circ \alpha = \{(f \circ \alpha)(\mathrm{sk}^*_1, \ldots, \mathrm{sk}^*_n) = f(\alpha(\mathrm{sk}^*_1), \ldots, \alpha(\mathrm{sk}^*_n)) : f \in \mathcal{F}\}$.*

We emphasize that in Theorem 1.1, we start with a scheme $\mathcal{E}$ that is entropy-k KDM$^{(n)}$-secure w.r.t. a function class $\mathcal{F}$, and we end up with a scheme $\mathcal{E}^*_\alpha$ that is not necessarily entropy-k secure anymore. However, it is KDM$^{(n)}$-secure w.r.t. a (supposedly richer) function class $\mathcal{F}'$ (see examples below). Therefore this theorem gives a way to convert security with weak keys, into enhanced KDM security. This gives a formal connection between the two notions.

We apply Theorem 1.1 to the schemes of [BHHO08] and [ACPS09] to obtain Theorems 1.2 and 1.3, respectively, presented below. In order to do that, we will argue that these schemes (or rather, a slight modification thereof) are entropy-k KDM$^{(1)}$-secure. In what follows, λ denotes the security parameter.

Theorem 1.2 (informal)**.** *Under the* DDH *assumption in a group $\mathbb{G}$ of order q, for any class $\mathcal{H} = \{h_1, \ldots, h_\ell : h_i \in \{0,1\}^k \to \{0,1\}\}$ of* poly(λ)*-time computable functions, with cardinality $\ell = \mathrm{poly}(\lambda)$, letting g be any generator of $\mathbb{G}$, there exists a* KDM$^{(1)}$*-secure encryption scheme w.r.t. the class of functions*

$$\mathcal{F}_\mathcal{H} = \left\{ f(g^{\mathbf{x}}) = g^{\sum_{i \in [\ell]} t_i h_i(\mathbf{x}) + w} : \mathbf{x} \in \{0,1\}^k, (\mathbf{t}, w) \in \mathbb{Z}_q^\ell \times \mathbb{Z}_q \right\} .$$

In this scheme, the secret-key is a vector in $\mathbb{G}^k$ whose i^{th} coordinate is $g^{x_i} \in \{1, g\}$. Theorem 1.2 is obtained by applying Theorem 1.1 to the [BHHO08] public-key encryption scheme, which is KDM secure w.r.t. affine functions in the exponent, using the mapping $\alpha(g^{\mathbf{x}}) = (g^{h_1(\mathbf{x})}, \ldots, g^{h_\ell(\mathbf{x})})$.

In particular, taking $\mathcal{H}$ to be the class of all degree-d monomials, we show that for any constant $d \in \mathbb{N}$, there exists a public-key encryption scheme that is KDM$^{(1)}$ secure w.r.t. all polynomials of total degree d (in the exponent). This is because degree-d polynomials over k variables can be viewed as affine functions applied to the vector of degree-d monomials. A different selection of $\mathcal{H}$ implies that for any polynomial t, there exists a public-key scheme that is KDM$^{(1)}$-secure w.r.t. all Turing machines of description length bounded by $\log t$ and running time bounded by t.

Theorem 1.3 (informal)**.** *Under the* LWE *assumption with modulus $q = p^2$, for a prime p, for any class $\mathcal{H} = \{h_1, \ldots, h_\ell : h_i \in \{0,1\}^k \to \{0,1\}\}$ of* poly(λ)*-time computable functions, with cardinality*

[1] This notion is different from security with key-leakage, where the leakage may depend on the public-key.

$\ell = \mathrm{poly}(\lambda)$, *there exists a* $\mathrm{KDM}^{(1)}$*-secure encryption scheme w.r.t. the class of functions*

$$\mathcal{F}_{\mathcal{H}} = \left\{ f(\mathbf{x}) = \sum_{i \in [\ell]} t_i h_i(\mathbf{x}) + w \pmod{p} : (\mathbf{t}, w) \in \mathbb{Z}_p^\ell \times \mathbb{Z}_p \right\} .$$

The secret key space in this scheme is $\{0,1\}^k$. The result is obtained by applying Theorem 1.1 to (a variant of) the [ACPS09] public-key encryption scheme, which is KDM secure w.r.t. affine functions, using the mapping $\alpha(\mathbf{x}) = (h_1(\mathbf{x}), \ldots, h_\ell(\mathbf{x}))$.

In a similar manner to the DDH based result, appropriate selections of $\mathcal{H}$ imply a $\mathrm{KDM}^{(1)}$ secure scheme w.r.t. all polynomials of total degree d and a $\mathrm{KDM}^{(1)}$-secure scheme w.r.t. all Turing machines of description length bounded by $\log t$ and running time bounded by t, for $t = \mathrm{poly}(\lambda)$.

We are able to extend the above results, using additional techniques (Theorem 1.1 will not suffice), and show that for the case of degree-d polynomials, both schemes obtained above are in fact $\mathrm{KDM}^{(n)}$ secure, based on their respective assumptions. These results are stated in the theorems below.

Theorem 1.4 (informal)**.** *Under the* DDH *assumption, for any* $d \in \mathbb{N}$*, there exists a public-key encryption scheme that is* $\mathrm{KDM}^{(n)}$*-secure w.r.t. degree-*d *polynomials in the exponent, for any* $n = \mathrm{poly}(\lambda)$.

Theorem 1.5 (informal)**.** *Under the* LWE *assumption, for any* $d \in \mathbb{N}$*, there exists a public-key encryption scheme that is* $\mathrm{KDM}^{(n)}$*-secure w.r.t. degree-*d *polynomials, for any* $n = \mathrm{poly}(\lambda)$.

Let us compare our results with the known impossibility results. As mentioned above, [HH09] showed a black-box impossibility result for obtaining $\mathrm{KDM}^{(1)}$-security w.r.t. the class of all functions, based on essentially any cryptographic assumption. This is done by showing that an adversary that breaks $\mathrm{KDM}^{(1)}$-security w.r.t. a random function cannot be useful in breaking the underlying assumption.

In their reduction, it is important that the random function, relative to which KDM-security is broken, is chosen *after* the encryption scheme has been defined. In contrast, in this work, the definition of the encryption scheme is dependent on the set of functions relative to which KDM security should hold. Namely, if the encryption scheme can evaluate the function, even in a black-box manner, then KDM-security w.r.t. that function can be obtained (based on cryptographic assumptions).

In other words, we show that even though it may be hard to achieve KDM security w.r.t. all possible functions, it is possible to do so w.r.t. any arbitrary, but small enough class of functions, i.e. one of polynomial cardinality.

Such a tool can be useful when, as a part of a cryptographic protocol, encryptions of certain functions of the secret-key need to be transmitted. In such case, one can tailor an encryption scheme to the required set of functions.

1.2 Our Techniques

Let us present the intuition behind the KDM amplification theorem (Theorem 1.1). Given an encryption scheme $\mathcal{E}$ that is entropy-k $\mathrm{KDM}^{(n)}$-secure w.r.t. a function class $\mathcal{F}$, we construct the encryption scheme $\mathcal{E}^*$ as follows: The key generation algorithm G^*, rather than choosing the secret key from $\mathcal{S}$, it chooses $\mathrm{sk} \stackrel{\$}{\leftarrow} \mathcal{K}$, and sets pk to be the public key corresponding to the secret key $\alpha(\mathrm{sk})$. As an example, one can think of $\mathcal{K} = \{0,1\}^k$, $\mathcal{S} = \{0,1\}^\ell$ where $\ell = \sum_{i=0}^{d} \binom{k}{i}$, and $\alpha(\mathrm{sk})$ is the vector of all monomials of degree d; namely, $\alpha(x_1, \ldots, x_k) = (\prod_{i \in I} x_i)_{|I| \le d}$, where $\mathrm{sk} = (x_1, \ldots, x_k) \in \{0,1\}^k$. Another example is where $\mathcal{K} = \{0,1\}^k$, $\mathcal{S} = \{0,1\}^{\mathrm{poly}(k)}$, and $\alpha(\mathrm{sk})$ as being the vector of all Turing machines with description length $O(\log k)$ and running time at most t (for some polynomial t), applied to sk. Namely, $\alpha(\mathrm{sk}) = \langle M(\mathrm{sk}) \rangle_M$, where M is a Turing machine with description length $O(\log k)$ that runs for at most t steps on sk.

The encryption algorithm E^* is identical to E. The decryption algorithm D^* takes the secret key sk, computes $\alpha(\mathrm{sk})$, and decrypts the ciphertext by applying the decryption algorithm D with the secret key $\alpha(\mathrm{sk})$.

We next exemplify why the scheme $\mathcal{E}^*$ has amplified KDM security. Assume, for example, that $\mathcal{E}$ was entropy-k $\mathrm{KDM}^{(1)}$ secure w.r.t. all affine functions. Consider, as in the example above, $\alpha(\mathrm{sk})$ that is the vector of all monomials of degree d. Then $\mathcal{E}^*$ is still secure, because it applies the scheme $\mathcal{E}$ with a weak secret key of min-entropy k. Moreover, the fact that $\mathcal{E}$ is entropy-k $\mathrm{KDM}^{(1)}$-secure w.r.t. all affine functions, implies that the scheme $\mathcal{E}^*$ is secure w.r.t. all affine functions of $\alpha(\mathrm{sk})$, i.e. all degree d polynomials of sk. Similarly, if $\alpha(\mathrm{sk})$ is the vector of all Turing machines with description length $O(\log k)$ with running time at most t, applied to sk, then $\mathcal{E}^*$ would be $\mathrm{KDM}^{(1)}$ secure w.r.t. all functions computed by these Turing machines.

Thus, Theorem 1.1 provides us with a generic tool that can be used to amplify KDM security of schemes that are entropy-k KDM-secure to begin with. However, the question that remains is: *Do there exist entropy-k KDM-secure schemes?*

$\mathrm{KDM}^{(1)}$**-security.** [BHHO08, ACPS09] presented encryption schemes that are $\mathrm{KDM}^{(1)}$-secure w.r.t. some classes of functions. We argue that these schemes are in fact entropy-k $\mathrm{KDM}^{(1)}$-secure (for some setting of parameters). This enables us to apply Theorem 1.1 and amplify $\mathrm{KDM}^{(1)}$-security "for free". Specifically, this implies $\mathrm{KDM}^{(1)}$-secure schemes w.r.t. degree-d polynomials or bounded description and bounded running time Turing machines.

$\mathrm{KDM}^{(n)}$**-security.** Two problems arise when trying to utilize Theorem 1.1 to obtain $\mathrm{KDM}^{(n)}$ security. First, a direct application of Theorem 1.1 may not produce the strongest result. Consider, for example, the case of bounded degree polynomials. Even if we had a scheme that was entropy-k $\mathrm{KDM}^{(n)}$-secure w.r.t. affine functions, Theorem 1.1 would only imply a scheme that is $\mathrm{KDM}^{(n)}$-secure w.r.t. bounded-degree polynomials where each monomial only contains variables of the same secret key. Second, we are not able to show entropy-k $\mathrm{KDM}^{(n)}$-security for any scheme and therefore cannot satisfy the conditions of the theorem.

To obtain Theorems 1.4 and 1.5, therefore, additional ideas are required. Rather than applying Theorem 1.1 directly for $\mathrm{KDM}^{(n)}$, we consider the schemes obtained by Theorems 1.2 and 1.3 for the specific case where $\mathcal{H}$ is the class of all degree-d monomials. We then show that these schemes are not only $\mathrm{KDM}^{(1)}$-secure w.r.t. degree-d polynomials, but are also $\mathrm{KDM}^{(n)}$-secure w.r.t. the same class. We emphasize that monomials can contain variables from all secret-keys in the system. This part contains the bulk of technical difficulty of this work.

While the proof for each scheme requires special treatment, the crux of the idea in both cases is similar. We use the "linear" behavior exhibited by both underlying schemes (in the DDH-based scheme, linearity is in the exponent) which enables the following form of homomorphism: starting from a single public key, that corresponds to a secret key sk, it is possible to generate a public key that corresponds to a linearly-related secret-key. This is done without knowing the original secret key sk, only the (linear) relation. This, however, is not enough: as it turns out (and as hinted by the intuition of Theorem 1.1 provided above), we need to apply this homomorphism on secret-keys whose coordinates are low-degree monomials. Therefore we cannot use arbitrary linear transformations to "switch" between secret keys. We solve this problem by presenting a class of linear transformations that do preserve the structure of the input secret-key.

1.3 Other Related Works and Notions

One can consider an "entropy-k" variant for any security measure for public-key encryption, analogously to our definition of entropy-k KDM-security; i.e., requiring that the scheme remains secure, in the relative measure, even when the secret-key is sampled from an arbitrary entropy-k distribution. This notion is incomparable to that of key-leakage resilience, defined by Akavia, Goldwasser and Vaikuntanathan [AGV09], and by Naor and Segev [NS09]. On the one hand, the notion of entropy-k security is weaker since imperfect randomness is only used to generate the secret-key, while the computation of the corresponding public-key uses perfect randomness. On the other hand, key-leakage resilience is weaker since it requires security to hold, with high probability, over *some* family of distributions, whereas entropy-k security requires security to hold for *all* high min-entropy distributions.

The idea of modifying the secret-key of a public-key encryption scheme in order to achieve additional properties has been used before. In the KDM-secure scheme of [BHHO08], binary vectors in the exponent of a group generator are used as secret-keys, instead of the more natural selection of vectors in $\mathbb{Z}_q$. This is done in order to achieve KDM security w.r.t. the desired function class. In [NS09], the secret-key distribution of the [BHHO08] scheme is again modified, this time using vectors of higher dimension than required, thus achieving security against key-leakage. The KDM-secure public-key scheme of [ACPS09] is very similar to that of [Reg05], with one of the changes being that the secret-key distribution is selected from a narrow Gaussian rather than being uniform. This is done, again, in order for KDM-security to apply w.r.t. the desired set of functions.

1.4 Paper Organization

We provide notation and standard definitions in Section 2, new definitions and tools used throughout the paper appear in Section 3. The KDM amplification theorem (Theorem 1.1) is formally restated and proven in Section 4, where examples of applying it to specific function classes are also provided. Sections 5 and 6 feature our DDH and LWE based constructions, respectively. Specifically, Theorems 1.2 and 1.4 are formally restated and proven in Section 5, while Theorems 1.3 and 1.5 are restated and proven in Section 6.

2 Notation and Definitions

We denote scalars in plain lowercase ($x \in \{0,1\}$), vectors in bold lowercase ($\mathbf{x} \in \{0,1\}^k$) and matrices in bold uppercase ($\mathbf{X} \in \{0,1\}^{k \times k}$). All vectors are column vectors by default, a row vector is denoted $\mathbf{x}^T$. The i^{th} coordinate of $\mathbf{x}$ is denoted x_i. For a set I, we use $\mathbf{x} = \langle x_i \rangle_{i \in I}$ to denote a vector that is indexed by elements in I.

Vectors in $\{0,1\}^k$ are treated both as elements in $\mathbb{Z}_q^k$ and as elements in $\mathbb{Z}_2^k$. We use standard arithmetic notation for arithmetics over $\mathbb{Z}_q^k$ and use $\mathbf{x} \oplus \mathbf{y}$ to denote the addition in $\mathbb{Z}_2^k$ (i.e. bitwise XOR operation).

For a group $\mathbb{G}$ with generator g and order q, if $\mathbf{x} \in \mathbb{Z}_q^n$ then $g^{\mathbf{x}} \in \mathbb{G}^n$ denotes the vector whose i^{th} coordinate is g^{x_i}; similarly we denote $g^{\mathbf{X}}$ for matrices. For sets $S \subseteq \mathbb{Z}_q$ we denote $g^S = \{g^x : x \in S\}$.

We note that given $\mathbf{X} \in \mathbb{Z}_q^{m\times n}$, $\mathbf{Y} \in \mathbb{Z}_q^{n\times k}$ it is possible to compute $g^{\mathbf{XY}}$ given either $(g^{\mathbf{X}}, \mathbf{Y})$ or $(\mathbf{X}, g^{\mathbf{Y}})$ using $\mathrm{poly}(m, n, k)$ group multiplications.

Let X be a probability distribution over domain S, we write $x \stackrel{\$}{\leftarrow} X$ to indicate that x is sampled from distribution X. X^n denotes the n-fold product distribution of X over S^n. The uniform distribution over a set S is denoted $U(S)$. We use $x \stackrel{\$}{\leftarrow} S$ as abbreviation for $x \stackrel{\$}{\leftarrow} U(S)$. The *min entropy* of a random variable X over domain S is $\mathbf{H}_\infty(X) = -\log\left(\max_{x\in S}\Pr[X = x]\right)$. Logarithms here, and anywhere else in this paper, are taken to the base 2. For any function f with domain S we let $f(X)$ denote the random variable (or corresponding distribution) obtained by sampling $x \stackrel{\$}{\leftarrow} X$ and outputting $f(x)$.

We write $\mathrm{negl}(n)$ to denote an arbitrary *negligible* function, i.e. one that vanishes faster than the inverse of any polynomial.

The *statistical distance* between two distributions X, Y (or random variables with those distributions) over common domain S is defined as $\mathsf{SD}(X, Y) = \max_{A\subseteq S}\left|\Pr[X \in A] - \Pr[Y \in A]\right|$. Two ensembles $\{X_n\}_n$, $\{Y_n\}_n$ are *statistically indistinguishable* if $\mathsf{SD}(X_n, Y_n) = \mathrm{negl}(n)$, and are *computationally indistinguishable* if for every $\mathrm{poly}(n)$-time adversary $\mathcal{A}$ it holds that

$$\left|\Pr[\mathcal{A}(X_n) = 1] - \Pr[\mathcal{A}(Y_n) = 1]\right| = \mathrm{negl}(n) \ .$$

Let M be a deterministic Turing Machine. We use $|M|$ to denote the description length of M and use $\mathsf{exec}(M, 1^t, x)$ to denote the content of M's output tape after running on x for t computation steps. Clearly $\mathsf{exec}(M, 1^t, x)$ is computable in time $\mathrm{poly}(|M|, t)$.

2.1 Cryptographic Assumptions

Decision Diffie-Hellman (DDH**).** Let $\mathbb{G}$ be a group of prime order q (in fact, we consider a family of groups parameterized by security parameter λ). The DDH assumption (on $\mathbb{G}$) is that the distributions (g, g^x, g^y, g^z) and (g, g^x, g^y, g^{xy}) are computationally indistinguishable, where g is a random generator for $\mathbb{G}$ and $x, y, z \stackrel{\$}{\leftarrow} \mathbb{Z}_q$.

Learning with errors (LWE**).** We use the *decisional* version of the LWE ([Reg05]) assumption. For any $m, n, q \in \mathbb{N}$ such that $q > 2$, all functions of the security parameter λ, and any probability distribution χ on $\mathbb{Z}_q$, the $\mathrm{LWE}_{m,n,q,\chi}$ assumption is that the distributions $(\mathbf{A}, \mathbf{As} + \mathbf{x})$ and $(\mathbf{A}, \mathbf{u})$ are computationally indistinguishable, where $\mathbf{A} \stackrel{\$}{\leftarrow} \mathbb{Z}_q^{m\times n}$, $\mathbf{s} \stackrel{\$}{\leftarrow} \mathbb{Z}_q^n$, $\mathbf{x} \stackrel{\$}{\leftarrow} \chi^m$, $\mathbf{u} \stackrel{\$}{\leftarrow} \mathbb{Z}_q^m$.

We remark that the *search* version of the assumption, where the challenge is to find $\mathbf{s}$, is equivalent to the decisional version, for prime q, under $\mathrm{poly}(q)$-time reductions. It is shown in [ACPS09] that this equivalence also holds for $q = p^e$, for integer constant e and prime p, provided that χ is a distribution over $\mathbb{Z}_q$ that produces an element in $\{-\frac{p-1}{2}, \ldots, \frac{p-1}{2}\}$ with all but negligible probability.

Worst-case to average-case reductions of [Reg05, Pei09] can be used to obtain a connection between LWE instances and worst case lattice problems, for some (Gaussian like) distribution χ.

2.2 KDM **Security**

A public-key encryption scheme $\mathcal{E} = (G, E, D)$ is defined by its key generation, encryption and decryption algorithms. The key generation algorithm G takes as input the unary vector 1^λ, where λ is called the *security parameter* of the scheme. All other parameters of the scheme are parameterized by λ. We let $\mathcal{S} = \{\mathcal{S}_\lambda\}$ denote the space of secret keys and $\mathcal{M} = \{\mathcal{M}_\lambda\}$ denote the message space of the encryption scheme. We refer the reader to [Gol04] for a formal definition of encryption schemes and their security.

In the scenario of key-dependent messages, we wish to model the case where functions of the secret key can be encrypted, and require that the resulting ciphertexts are indistinguishable from encryptions of 0. We want our definition to apply also for the case of "key cycles" where a function of one user's secret key is encrypted by another's public key and vice versa. The most inclusive definition, therefore, is parameterized by the number of users n and allows encrypting a function of the entire vector of n secret keys under any of the corresponding public keys (this is sometimes referred to as "clique security"). An additional parameter to be considered is the set of functions of the secret key that we allow to encrypt. We use the definition presented in [BHHO08].

Formally, let $\mathcal{E} = (G, E, D)$ be a public key encryption scheme, $n > 0$ be an integer, $\mathcal{S} = \{\mathcal{S}_\lambda\}$ be the space of secret keys, and let $\mathcal{F} = \{\mathcal{F}_\lambda\}$ be a class of functions such that $\mathcal{F}_\lambda \subseteq \mathcal{S}_\lambda^n \to \mathcal{M}_\lambda$.

We define the $\mathrm{KDM}^{(n)}$ game, w.r.t. the function class $\mathcal{F}$, played between a challenger and an adversary $\mathcal{A}$, as follows.

Initialize. The challenger selects $b \xleftarrow{\$} \{0, 1\}$ and generates, for all $i \in [n]$, key pairs $(\mathrm{sk}_i, \mathrm{pk}_i) \xleftarrow{\$} G(1^\lambda)$. The challenger then sends $\{\mathrm{pk}_i\}_{i\in[n]}$ to $\mathcal{A}$.

Query. The adversary makes queries of the form $(i, f) \in [n] \times \mathcal{F}_\lambda$. For each query, the challenger computes $y \leftarrow f(\mathrm{sk}_1, \ldots, \mathrm{sk}_n)$ and sends the following ciphertext to $\mathcal{A}$.

$$c \leftarrow \begin{cases} E_{\mathrm{pk}_i}(y) & \text{if } b = 0 \\ E_{\mathrm{pk}_i}(0) & \text{if } b = 1. \end{cases}$$

Finish. $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$.

Adversary $\mathcal{A}$ *wins* the game if $b' = b$. The *advantage* of $\mathcal{A}$, denoted $\mathrm{KDM}^{(n)}\mathrm{Adv}[\mathcal{A}, \mathcal{E}](\lambda)$ is $|\Pr[W] - 1/2|$ where W is the event that $\mathcal{A}$ wins.

We sometime denote $\mathrm{KDM}^{(n)}_{\mathcal{F}}$ to indicate the function class in discussion.

3 New Definitions and Tools

3.1 Projective Encryption Schemes and Weak Keys

Projection. Throughout this paper, we only consider encryption schemes that have a *projection* between the secret and public key. Namely, the key generation can be described as first sampling the secret key from some set and then applying an efficiently computable projection function (which can be randomized) to generate the public key.

Definition 3.1 (projection). *Let $\mathcal{E} = (G, E, D)$ be a public-key encryption scheme. $\mathcal{E}$ is* projective *if $G(1^\lambda) = (\mathrm{sk}, \mathrm{pk} = \mathsf{Proj}(\mathrm{sk}))$ where $\mathrm{sk} \xleftarrow{\$} \mathcal{S}$ and $\mathsf{Proj}(\cdot)$ is an efficiently computable (possibly randomized) function.*

We remark that many known encryption schemes are indeed projective, e.g. [RSA78, Gam84, Reg05, BHHO08, ACPS09] and others. We further remark that any secure scheme can be modified to be projective by using the randomness of the key generation as the secret key. However such transformation does not preserve KDM security (formally defined below) and thus we will need to require projection explicitly.

Weak keys and entropy-k security. We are also interested in a more specific case where a (projective) scheme remains secure even when the key generation is "improper": the secret key is sampled from an arbitrary distribution on $\mathcal{S}$ that has min-entropy k. The projection is then applied to the sampled value.

We can think of an "entropy-k variant" of any security notion σ, we thus provide a general definition. In this work, however, we instantiate this definition with σ being KDM security.

Definition 3.2 (entropy-k security)**.** *Let $\mathcal{E} = (G, E, D)$ be a projective public-key encryption scheme and let σ be some security notion. Consider a distribution ensemble $\mathcal{D} = \{\mathcal{D}_\lambda\}$ over $\mathcal{S} = \{\mathcal{S}_\lambda\}$. Let $G_\mathcal{D}$ denote the following key-generator: $G_\mathcal{D}(1^\lambda) = (\mathrm{sk}, \mathsf{Proj}(\mathrm{sk}))$ where $\mathrm{sk} \leftarrow \mathcal{D}_\lambda$.*

Let $k : \mathbb{N} \to \mathbb{R}^+$ be some function. $\mathcal{E}$ is entropy-k σ-secure *if for any ensemble $\mathcal{D}$ with $\mathbf{H}_\infty(\mathcal{D}_\lambda) \geq k(\lambda)$ it holds that $\mathcal{E}_\mathcal{D}(G_\mathcal{D}, E, D)$ is σ-secure.*

We stress that entropy-k security, as defined above, is a notion incomparable to that of key-leakage resilience (as defined in [AGV09, NS09]). On the one hand, the notion of entropy-k security is weaker since imperfect randomness is only used to generate the secret-key, while the projection $\mathsf{Proj}(\cdot)$ uses perfect randomness to compute the corresponding public-key. On the other hand, key-leakage resilience is weaker since it requires security to hold with high probability over *some* family of distributions, whereas entropy-k security requires security to hold for *all* high min-entropy distributions.

3.2 Transformations on Expanded Secret-Keys

Let q be some modulus. The set of *affine functions* modulo q on $\mathbb{Z}_q^k$ is

$$\mathcal{F}_{\text{aff}} = \{f_{\mathbf{t},w}(\mathbf{x}) = \mathbf{t}^T\mathbf{x} + w : (\mathbf{t}, w) \in \mathbb{Z}_q^k \times \mathbb{Z}_q\} \ .$$

The set of *affine functions in the exponent* over $\mathbb{G}^k$, where $\mathbb{G}$ is a group of order q and g is a generator of $\mathbb{G}$, is denoted by

$$\hat{\mathcal{F}}_{\text{aff}} = \{h_{\mathbf{t},w}(g^{\mathbf{x}}) = g^{\mathbf{t}^T\mathbf{x}+w} : (\mathbf{t}, w) \in \mathbb{Z}_q^k \times \mathbb{Z}_q\} \ .$$

Degree-d polynomials over k variables can be viewed as affine functions applied to the vector of degree-d monomials. While we consider polynomials over $\mathbb{Z}_q$, we only apply them to binary variables, $\mathbf{x} \in \{0,1\}^k$. We define a mapping $\boldsymbol{\gamma}_{k,d}$ that maps $\mathbf{x} \in \{0,1\}^k$ into the vector containing all monomials of degree d of the variables of $\mathbf{x}$.

Definition 3.3 (the vector of monomials $\boldsymbol{\gamma}_{k,d}$)**.** *For all $k, d \in \mathbb{N}$ and $\mathbf{x} \in \{0,1\}^k$, we define the vector of all degree-d monomials in $\mathbf{x}$ by*

$$\boldsymbol{\gamma}_{k,d}(\mathbf{x}) = \left\langle \prod_{j \in J} x_j \right\rangle_{\substack{J \subseteq [k], \\ |J| \leq d}} \ .$$

In other words, letting $\nu_{k,d} = \sum_{j=0}^{d} \binom{k}{j}$ denote the number of such degree-d monomials, $\boldsymbol{\gamma}_{k,d} : \{0,1\}^k \to \{0,1\}^{\nu_{k,d}}$ is a mapping between vectors. We denote its image by $\Gamma_{k,d} = \left\{\boldsymbol{\gamma}_{k,d}(\mathbf{x}) : \mathbf{x} \in \{0,1\}^k\right\}$.

It follows immediately from the definition that $\boldsymbol{\gamma}_{k,d}$ is injective, since $(\boldsymbol{\gamma}_{k,d}(\mathbf{x}))_{\{i\}} = x_i$, and thus that $|\Gamma_{k,d}| = 2^k$.

Intuitively, in the context of KDM-security amplification, $\mathbf{x}$ is our "real" secret-key, whereas $\boldsymbol{\gamma}_{k,d}(\mathbf{x})$, the expanded version of $\mathbf{x}$, is used as a "secret-key" for a scheme that is KDM-secure w.r.t. affine functions. This results in a KDM-secure scheme w.r.t. degree-d polynomials.

We denote the set of all *degree-d polynomials* over $\mathbb{Z}_q$ with binary variables $\mathbf{x} \in \{0,1\}^k$ by

$$\mathcal{F}_d = \{f_{\mathbf{t}}(\mathbf{x}) = \mathbf{t}^T \cdot \boldsymbol{\gamma}_{k,d}(\mathbf{x}) : \mathbf{t} \in \mathbb{Z}_q^\ell\} \ .$$

Note that $\boldsymbol{\gamma}_{k,d}(\mathbf{x})_\emptyset = 1$, i.e. the vector of monomials contains the empty monomial that always evaluates to 1. Therefore there is no need for an additional free term w as in the definition of affine functions.

Again, for the *degree-d polynomials in exponent* we denote

$$\hat{\mathcal{F}}_d = \{h_{\mathbf{t}}(g^{\mathbf{x}}) = g^{\mathbf{t}^T \cdot \boldsymbol{\gamma}_{k,d}(\mathbf{x})} : \mathbf{t} \in \mathbb{Z}_q^\ell\} \ ,$$

where g is a generator of a group $\mathbb{G}$ of order q.

The following lemma states that that given $\mathbf{y} \in \{0,1\}^k$, we can efficiently compute a matrix $\mathbf{T} \in \mathbb{Z}_q^{\ell\times\ell}$ such that for all $\mathbf{x} \in \{0,1\}^k$ it holds that $\mathbf{T} \cdot \boldsymbol{\gamma}_{k,d}(\mathbf{x}) = \boldsymbol{\gamma}_{k,d}(\mathbf{x} \oplus \mathbf{y})$. We think of $\mathbf{y}$ as the known relation between secret-keys $\mathbf{x}$ and $\mathbf{x} \oplus \mathbf{y}$. The transformation $\mathbf{T}$ allows us to convert the expanded version of $\mathbf{x}$ to the expanded version of $\mathbf{x} \oplus \mathbf{y}$, i.e. to convert $\boldsymbol{\gamma}_{k,d}(\mathbf{x})$ into $\boldsymbol{\gamma}_{k,d}(\mathbf{x} \oplus \mathbf{y})$.

Lemma 3.1. *For all $k, d, q \in \mathbb{N}$ such that $q > 2$, there exists an efficiently computable function $\mathsf{T}_{k,d,q} : \{0,1\}^k \to \mathbb{Z}_q^{\ell\times\ell}$, where $\ell = \nu_{k,d}$, such that setting $\mathbf{T} = \mathsf{T}_{k,d,q}(\mathbf{y})$, for all $\mathbf{x} \in \{0,1\}^k$ it holds that $\mathbf{T} \cdot \boldsymbol{\gamma}_{k,d}(\mathbf{x}) = \boldsymbol{\gamma}_{k,d}(\mathbf{x} \oplus \mathbf{y})$. Moreover $\mathbf{T}$ is an involution, i.e. $\mathbf{T}^2$ is the identity matrix.*

Proof. Fix k, d, q, ℓ and $\mathbf{y} \in \{0,1\}^k$. For any $\mathbf{x} \in \{0,1\}^k$ it holds that

$$(\mathbf{x} \oplus \mathbf{y})_i = \begin{cases} x_i & y_i = 0 \\ 1 - x_i & y_i = 1 \end{cases}$$

where the arithmetics is over $\mathbb{Z}_q$. Hence, given $\mathbf{y}$, we can compute $\mathbf{v}, \mathbf{w} \in \mathbb{Z}_q^k$ such that $(\mathbf{x} \oplus \mathbf{y})_i = v_i x_i + w_i$: if $y_i = 0$ then $v_i = 1, w_i = 0$ and if $y_i = 1$ then $v_i = -1, w_i = 1$. Thus, for all $J \subseteq [k]$, $|J| \le d$, it holds that $\boldsymbol{\gamma}_{k,d}(\mathbf{x} \oplus \mathbf{y})_J = \prod_{i\in J}(v_i x_i + w_i)$. We can now open the parenthesis of the expression (note that this can be done in time $\mathrm{poly}(\ell)$) and express $\boldsymbol{\gamma}_{k,d}(\mathbf{x} \oplus \mathbf{y})_J$ as a degree-d polynomial in $\mathbf{x}$ with known coefficients, or, in other words, as a linear function of $\boldsymbol{\gamma}_{k,d}(\mathbf{x})$. These coefficients will constitute the J^{th} row of the matrix $\mathbf{T} = \mathsf{T}(\mathbf{y})$. Computing row by row, we can construct a matrix $\mathbf{T}$ such that $\mathbf{T}\boldsymbol{\gamma}_{k,d}(\mathbf{x}) = \boldsymbol{\gamma}_{k,d}(\mathbf{x} \oplus \mathbf{y})$ as desired.

We note that $\mathbf{T}^2 \cdot \boldsymbol{\gamma}_{k,d}(\mathbf{x}) = \mathbf{T} \cdot \boldsymbol{\gamma}_{k,d}(\mathbf{x} \oplus \mathbf{y}) = \boldsymbol{\gamma}_{k,d}(\mathbf{x})$ and thus conclude that $\mathbf{T}^2$ is the identity matrix. In order to derive this last conclusion, we rely on the fact that there exist ℓ linearly-independent vectors of the form $\boldsymbol{\gamma}_{k,d}(\mathbf{x})$. □

4 Amplification of KDM Security

In this section we give a general result: We show that an entropy-k KDM-secure scheme, w.r.t. a certain class of functions, can be converted into various schemes that are KDM-secure w.r.t. richer classes. We start by stating the general result and then present corollaries for specific classes of functions.

4.1 Main Theorem

Before stating our theorem, let us give some intuition for how KDM-security can be amplified for projective entropy-k schemes (as defined in Section 3.1).

Consider, for example, a projective encryption scheme $\mathcal{E}$ that is entropy-k KDM-secure w.r.t. the class of indexing functions $\mathcal{I} = \{h_i(\mathbf{s}) = s_i\}$ or, in other words, a bit by bit encryption of the secret key is secure. Entropy-k security in particular means that we can sample the secret key $\mathrm{sk} = \mathbf{s} \in \{0,1\}^\ell$ as follows: first, sample the first k bits uniformly, call this part $\mathbf{x}$; then, set the remaining bits of $\mathbf{s}$ to $s_i = f_i(\mathbf{x})$, where $\{f_i\}_{i=k+1,\ldots,\ell}$ is an arbitrary class of efficiently computable deterministic functions. The resulting secret-key distribution has min-entropy k and thus $\mathcal{E}$ is still KDM-secure w.r.t. $\mathcal{I}$ with the resulting secret-key distribution. Namely, $\mathcal{E}$ is secure w.r.t. the functions $h_i(\mathbf{s}) = s_i = f_i(\mathbf{x})$. Therefore, we can convert $\mathcal{E}$ into a scheme $\mathcal{E}^*$ by setting the secret key in $\mathcal{E}^*$ to be $\mathbf{x}$. This $\mathcal{E}^*$ is KDM-secure w.r.t. indexing functions as well as the functions $\{f_i\}_{i=k+1,\ldots,\ell}$.

Theorem 1.1 (restated)**.** *Let $\mathcal{E} = (G, E, D)$ be a projective public-key encryption scheme that is entropy-k $\mathrm{KDM}^{(n)}$-secure w.r.t. a function class $\mathcal{F}$. Let $\mathcal{S} = \{\mathcal{S}_\lambda\}$ be the space of secret keys.*

Let $\mathcal{K} = \{\mathcal{K}_\lambda\}$ be a family of sets such that $|\mathcal{K}| \geq 2^k$ and let $\alpha : \mathcal{K} \to \mathcal{S}$ be a deterministic, efficiently computable and injective function. Then there exists a projective encryption scheme $\mathcal{E}^*_\alpha = (G^*, E^*, D^*)$ with secret-key space $\mathcal{K}$ that is $\mathrm{KDM}^{(n)}$ secure w.r.t. $\mathcal{F} \circ \alpha = \{(f \circ \alpha)(\mathrm{sk}_1, \ldots, \mathrm{sk}_n) = f(\alpha(\mathrm{sk}_1), \ldots, \alpha(\mathrm{sk}_n)) : f \in \mathcal{F}\}$.

Proof. Consider the ensemble $\mathcal{D}$ where $\mathcal{D}_\lambda = \alpha(U(\mathcal{K}_\lambda))$ and consider the scheme $\mathcal{E}_\mathcal{D} = (G_\mathcal{D}, E, D)$ as in Definition 3.2. $\mathcal{E}^*_\alpha$ is similar to $\mathcal{E}_\mathcal{D}$ with the following modifications. $G^*(1^\lambda)$ first samples $\mathrm{sk}^* \xleftarrow{\$} \mathcal{K}$ and then computes $\mathrm{pk} = \mathsf{Proj}^*(\mathrm{sk}^*) = \mathsf{Proj}(\alpha(\mathrm{sk}^*))$. Note that the distribution of the public-keys is identical to that of $\mathcal{E}_\mathcal{D}$ while the distributions of secret-keys differ. The encryption E^* is performed identically to E. The decryption $D^*_{\mathrm{sk}^*}(c)$ is performed by first computing $\mathrm{sk} = \alpha(\mathrm{sk}^*)$ and then outputting $D_{\mathrm{sk}}(c)$.

Since α is injective, it holds that $\mathbf{H}_\infty(\mathcal{D}_\lambda) \geq k$, and thus by definition, $\mathcal{E}_\mathcal{D}$ is $\mathrm{KDM}^{(n)}$-secure w.r.t. $\mathcal{F}$.

We next show that for any adversary $\mathcal{A}^*$ for the $\mathrm{KDM}^{(n)}$ game with $\mathcal{E}^*_\alpha$, there exists an adversary $\mathcal{A}$ for the $\mathrm{KDM}^{(n)}$ game with $\mathcal{E}_\mathcal{D}$ such that

$$\mathrm{KDM}^{(n)}_{\mathcal{F}}\mathrm{Adv}[\mathcal{A}, \mathcal{E}_\mathcal{D}](\lambda) = \mathrm{KDM}^{(n)}_{\mathcal{F}\circ\alpha}\mathrm{Adv}[\mathcal{A}^*, \mathcal{E}^*_\alpha](\lambda) \ .$$

This will complete the proof of the theorem.

Adversary $\mathcal{A}$ simulates $\mathcal{A}^*$.

Initialize. Since the public key distributions of $\mathcal{E}_\mathcal{D}$ and $\mathcal{E}^*_\alpha$ are identical, $\mathcal{A}$ forwards its input $\mathrm{pk}_1, \ldots, \mathrm{pk}_n$ to $\mathcal{A}^*$.

Queries. When $\mathcal{A}^*$ sends the query $(i, f \circ \alpha) \in [n] \times (\mathcal{F} \circ \alpha)$, $\mathcal{A}$ sends the query (i, f).[2] Let sk^*_i denote the secret-key corresponding to pk_i in $\mathcal{E}^*_\alpha$, then by definition $\mathrm{sk}_i = \alpha(\mathrm{sk}^*_i)$ is the secret-key corresponding to pk_i in $\mathcal{E}_\mathcal{D}$. Therefore $f(\mathrm{sk}_1, \ldots, \mathrm{sk}_n) = (f \circ \alpha)(\mathrm{sk}^*_1, \ldots, \mathrm{sk}^*_n)$, and $\mathcal{A}$ can forward the answer to $\mathcal{A}^*$. Thus, $\mathcal{A}$ can simulate any query made by $\mathcal{A}^*$ during the game.

Finish. When $\mathcal{A}^*$ terminates and returns b', $\mathcal{A}$ also terminates and returns the same b'.

Since $\mathcal{A}$ simulates $\mathcal{A}^*$ exactly, it follows that $\mathcal{A}$ achieves the same advantage in the $\mathrm{KDM}^{(n)}$ game with $\mathcal{E}_\mathcal{D}$ as $\mathcal{A}^*$ does with $\mathcal{E}^*_\alpha$. □

4.2 Exemplifying for Specific Function Classes

We demonstrate specific cases where Theorem 1.1 amplifies KDM security. We restrict our attention to $\mathrm{KDM}^{(1)}$ security (see discussion below).

[2]We represent $f \circ \alpha$ in such a way that enables to derive f.

- *Bounded description functions.* We first show how to amplify the class of indexing functions $\mathcal{I} = \{h_i(\mathbf{s}) = s_i\}$ into the class of all functions computable by a Turing machine with bounded description length and bounded running time. Let $\mathcal{E}$ be an entropy-k KDM$^{(1)}$-secure encryption scheme w.r.t. the class of indexing functions, with message space $\mathcal{M} = \{0,1\}$ and secret-key space $\mathcal{S} = \{0,1\}^\ell$. Let $\mathcal{K} = \{0,1\}^k$ and $\alpha(\mathbf{x}) = \langle \mathsf{exec}(M, 1^{t(\lambda)}, \mathbf{x})\rangle_{|M| \le \log \ell}$ where $t(\cdot)$ is some (fixed) polynomial. Then $\mathcal{E}^*_\alpha$, defined in the proof of Theorem 1.1, is KDM$^{(1)}$-secure w.r.t. all functions computable by a Turing machine with description length $\log \ell$ and running time $t(\lambda)$.[3]

- *Bounded degree polynomials.* We now show how to amplify the class of affine functions into the class of bounded degree polynomials. Let $\mathcal{E}$ be an entropy-k KDM$^{(1)}$-secure encryption scheme w.r.t. the class of affine functions $\mathbb{F}^\ell \to \mathbb{F}$, with $\mathcal{M} = \mathbb{F}$ and $\mathcal{S} \subseteq \mathbb{F}^\ell$, for a finite field $\mathbb{F}$. Let $\mathcal{K} = \{0,1\}^k \subseteq \mathbb{F}^k$ and $\alpha(\mathbf{x}) = \boldsymbol{\gamma}_{k,d}(\mathbf{x})$ with $\boldsymbol{\gamma}_{k,d}$ as in Definition 3.3 (we assume that $\ell = \nu_{k,d}$). Namely, α contains all degree d monomials. Then $\mathcal{E}^*_\alpha$, defined in the proof of Theorem 1.1, is KDM$^{(1)}$-secure w.r.t. all degree-d polynomials $\mathbb{F}^k \to \mathbb{F}$.

We provided examples only for the case of KDM$^{(1)}$-security for two reasons. First of all, while we present in Sections 5.2, 6.2 we present (candidates for) entropy-k KDM$^{(1)}$-secure schemes, we are unable to obtain entropy-k KDM$^{(n)}$-secure schemes for $n > 1$. Secondly, even if such exist, the result of applying Theorem 1.1 for the classes above would be weaker than expected. This is because while the functions in the class $\mathcal{F}$ are applied to the vector of n secret keys, the mapping α is only applied to one secret-key at a time. Therefore, the first example above would imply KDM$^{(n)}$-security w.r.t. Turing machines that only take one of the secret keys as input; the second would imply KDM$^{(n)}$-security w.r.t. degree-d polynomials where each monomial only contains variables from one secret key.

5 DDH Based KDM Security

For any constant d, we present a scheme that is KDM$^{(n)}$ secure w.r.t. all degree-d polynomials (in the exponent), $\hat{\mathcal{F}}_d$. We also present a scheme that is KDM$^{(1)}$-secure w.r.t. the class of all functions computed by Turing machines with description length at most $\log t$ and running time t, for some polynomial t (more generally, w.r.t. any class of efficiently computable functions of polynomial cardinality). Our starting point is the scheme presented in [BHHO08], which we denote $\mathcal{E}_{\text{BHHO}}$, which is extended using ideas from Section 4.

In Section 5.1, we present $\mathcal{E}_{\text{BHHO}}$ and state its entropy-k KDM-security properties. Then, in Section 5.2, we show how to use Theorem 1.1 to amplify the KDM$^{(1)}$-security of the scheme to richer classes of functions, including $\hat{\mathcal{F}}_d$. Finally, in Section 5.3, we show that the KDM$^{(1)}_{\hat{\mathcal{F}}_d}$-secure scheme is also KDM$^{(n)}_{\hat{\mathcal{F}}_d}$-secure.

5.1 Scheme $\mathcal{E}_{\text{BHHO}}$

The scheme, as defined in [BHHO08], assumes that the secret-key is sampled uniformly from $g^{\mathcal{S}}$ for a specific set $\mathcal{S} = \{0,1\}^\ell$. They discussed the possibility of using different sets $\mathcal{S}$ in the context

[3] One has to be careful when showing that α is injective. We can either assume that the first k coordinates of the output contain the input, or, if ℓ is sufficiently larger than k, we can rely on the short description and running time of the indexing functions.

of improving efficiency. For our purposes, we take $\mathcal{S}$ as one of the parameters of the scheme. The scheme $\mathcal{E}_{\text{BHHO}}[\mathbb{G},\mathcal{S}]$ is defined as follows.

Parameters. Let $\mathbb{G}$ be a group of order q such that $\log q = \text{poly}(\lambda)$ and let g be some generator of $\mathbb{G}$. Let $\ell = \text{poly}(\lambda)$ and $\mathcal{S} \subseteq \mathbb{Z}_q^\ell$. We require that group operations over $\mathbb{G}$ can be done efficiently (in time $\text{poly}(\lambda)$). The secret-key space of the scheme is $g^{\mathcal{S}}$ and the message space is $\mathbb{G}$. We require that $\mathcal{S}$ is such that there exists an efficiently computable mapping that, for all $\mathbf{s} \in \mathcal{S}$, takes $g^{\mathbf{s}}$ and returns $\mathbf{s}$.

Key generation. On input 1^λ, the generator samples $\mathbf{s} \xleftarrow{\$} \mathcal{S}$ and sets the secret-key $\text{sk} = g^{\mathbf{s}} \in \mathbb{G}^\ell$. It then samples $\mathbf{z} \xleftarrow{\$} \mathbb{Z}_q^\ell$ and sets the public-key $\text{pk} = (g^{\mathbf{z}}, g^{-\mathbf{z}^T\cdot\mathbf{s}}) \in \mathbb{G}^\ell \times \mathbb{G}$.

Encryption. On inputs a public-key $\text{pk} = (g^{\mathbf{z}}, g^v) \in \mathbb{G}^\ell \times \mathbb{G}$, and a message $w \in \mathbb{G}$, encryption is done by sampling $r \xleftarrow{\$} \mathbb{Z}_q$ and outputting $(g^{r\cdot\mathbf{z}}, g^{r\cdot v}\cdot w)$.

Decryption. On inputs a secret-key $\text{sk} = g^{\mathbf{s}}$ and a ciphertext $\mathbf{c} = (g^{\mathbf{a}}, g^u)$, the decryption process is as follows. First $\mathbf{s}$ is extracted from sk (note that we define $\mathcal{S}$ so that this can be done efficiently) and output $w = g^{\mathbf{s}^T\cdot\mathbf{a}} \cdot g^u$.

The following statement on the security of $\mathcal{E}_{\text{BHHO}}[\mathbb{G},\mathcal{S}]$ is implicit in [BHHO08]. Specifically see Corollary 1 and the discussion in Section 4 in their work.

Lemma 5.1 ([BHHO08])**.** *If* $\mathsf{SD}\left((\mathbf{a}, \mathbf{a}^T\cdot\mathbf{s}), (\mathbf{a}, u)\right) = \text{negl}(\lambda)$ *for* $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_q^\ell$, $\mathbf{s} \xleftarrow{\$} \mathcal{S}$, $u \xleftarrow{\$} \mathbb{Z}_q$ *and if the* DDH *assumption holds for* $\mathbb{G}$, *then* $\mathcal{E}_{BHHO}[\mathbb{G},\mathcal{S}]$ *is* $\text{KDM}^{(1)}$ *secure w.r.t.* $\hat{\mathcal{F}}_{\mathit{aff}}$.

A useful corollary follows.

Corollary 5.2. $\mathcal{E}_{BHHO}[\mathbb{G},\mathcal{S}]$ *is entropy-k* $\text{KDM}^{(1)}$*-secure w.r.t.* $\hat{\mathcal{F}}_{\mathit{aff}}$ *if* $\mathcal{S} = \{0,1\}^\ell$, $q\cdot 2^{-k} = \text{negl}(\lambda)$ *and the* DDH *assumption holds.*

Proof. Consider $\mathcal{E}_{\text{BHHO}}[\mathbb{G},\mathcal{S}]$ where the DDH assumption holds in $\mathbb{G}$ and where $\mathcal{S} \subseteq \{0,1\}^\ell$ and $|\mathcal{S}| \geq 2^k$. In such case, there exists an efficiently computable mapping restoring $\mathbf{s} \in \mathcal{S}$ from $g^{\mathbf{s}}$, since $g^{s_i} \in \{1, g\}$.

In addition, an immediate corollary of the left-over hash lemma (see [BHHO08, Lemma 2]) implies that $\mathsf{SD}\left((\mathbf{a}, \mathbf{a}^T\cdot\mathbf{s}), (\mathbf{a}, u)\right) \leq \sqrt{q/(4\,|\mathcal{S}|)}$. Therefore, if $|\mathcal{S}| \geq 2^k$ where $q\cdot 2^{-k} = \text{negl}(\lambda)$, then Lemma 5.1 implies $\text{KDM}^{(1)}$-security of $\mathcal{E}_{\text{BHHO}}[\mathbb{G},\mathcal{S}]$.

Since the above holds for any $\mathcal{S}$ with $|\mathcal{S}| \geq 2^k$, entropy-k $\text{KDM}^{(1)}$-security follows. ☐

5.2 Amplification of $\text{KDM}^{(1)}$-Security

We use Theorem 1.1 and Corollary 5.2 to amplify the $\text{KDM}^{(1)}$-security of $\mathcal{E}_{\text{BHHO}}$. We say that a finite set of functions, $\mathcal{H} = \{h_1, \ldots, h_\ell\}$, with a common domain, is *entropy preserving* if $\alpha_{\mathcal{H}}(x) = (h_1(x), \cdots h_\ell(x))$ is an injective function.

Theorem 1.2 (restated)**.** *Let* $\mathbb{G}$ *be a group of order* q *for which the* DDH *assumption holds (more precisely: a family of groups parameterized by* λ*). Let* g *be any generator of* $\mathbb{G}$. *Let* k *be such that* $q\cdot 2^{-k} = \text{negl}(\lambda)$. *Let* $\mathcal{H} = \{h_1, \ldots, h_\ell : h_i \in \{0,1\}^k \to \{0,1\}\}$ *be an entropy preserving class of efficiently computable functions with cardinality* $\ell = \text{poly}(\lambda)$. *Then there exists a* $\text{KDM}^{(1)}$*-secure public-key encryption scheme w.r.t. the class of functions*

$$\mathcal{F}_{\mathcal{H}} = \left\{ f(g^{\mathbf{x}}) = g^{\sum_{i\in[\ell]} t_i h_i(\mathbf{x}) + w} : (\mathbf{t}, w) \in \mathbb{Z}_q^\ell \times \mathbb{Z}_q \right\} .$$

Proof. By Corollary 5.2, $\mathcal{E}_{\text{BHHO}}[\mathbb{G}, \{0,1\}^\ell]$ is entropy-k KDM$^{(1)}$-secure w.r.t. $\hat{\mathcal{F}}_{\text{aff}}$. We apply Theorem 1.1 to this scheme with $\alpha : g^{\{0,1\}^k} \to g^{\{0,1\}^\ell}$, where $\alpha(g^{\mathbf{x}}) = \left\langle g^{h(\mathbf{x})} \right\rangle_{h \in \mathcal{H}}$. To do this, we need to show that α is injective and efficiently computable: α is injective since $\mathcal{H}$ is entropy preserving and since g is a generator; moreover, it is efficiently computable since $\mathcal{H}$ is efficiently computable and since $\mathbf{x} \in \{0,1\}^k$, which means that it can be efficiently extracted from $g^{\mathbf{x}}$. Applying Theorem 1.1, there exists a KDM$^{(1)}$-secure scheme w.r.t. $\hat{\mathcal{F}}_{\text{aff}} \circ \alpha$. By definition of $\hat{\mathcal{F}}_{\text{aff}}$ and α, it holds that $\hat{\mathcal{F}}_{\text{aff}} \circ \alpha = \{f_{\mathbf{t},w}(g^{\mathbf{x}}) = g^{\sum_{h \in \mathcal{H}} t_i h(\mathbf{x}) + w} : (\mathbf{t}, w) \in \mathbb{Z}_q^\ell \times \mathbb{Z}_q\} = \mathcal{F}_{\mathcal{H}}$, as required. $\square$

The above calls for a short discussion on parameters. The "standard" form of the DDH assumption considers polylog(q)-time adversaries. Since we consider adversaries that run in time poly(λ), this implies that $\log q \geq \lambda^\epsilon$ for some $\epsilon > 0$. Corollary 5.2 requires that $q \cdot 2^{-k} = \text{negl}(\lambda)$, i.e. that $k \geq \log q + \omega(\log \lambda) \geq \lambda^\epsilon$. Therefore, if we base security on the standard DDH assumption then, since $\ell = \text{poly}(\lambda)$, it holds that $\ell = \text{poly}(k)$. This restricts the size of classes $\mathcal{H}$ for which we can apply Theorem 1.2. One example is letting $t = \text{poly}(\lambda)$ be some polynomial and taking $\mathcal{H}$ be the set of all functions computable by a Turing machine with description $\log \ell$ and running time at most t. In this case, $\ell = \text{poly}(k)$ means that we are restricted to Turing machines with description length at most $O(\log k)$. Another important example, discussed in detail below, is taking $\mathcal{H}$ to be the class of all monomials of degree-d. Here, the restriction $\ell = \text{poly}(k)$ means that we can only do so for $d = O(1)$.

We note, however, that if we make a stronger assumption, e.g. assume that the DDH assumption holds also for adversaries that run in time poly($2^{\log^\delta q}$), for some $\delta \in (0,1)$, then we could take $q = 2^{\log^{1/\delta} \lambda}$ and have $k = \log q + \omega(\log \lambda) = O(\log^{1/\delta} \lambda)$, i.e. $\ell = \text{poly}(\lambda) = 2^{\Omega(k^\delta)}$. In the example of degree-d monomials, since $\ell \leq (k+1)^d$, we can set $d = \frac{\log \ell}{\log(k+1)} = \tilde{\Omega}(k^\delta)$.

Recall that $\boldsymbol{\gamma}_{k,d}$, $\nu_{k,d}$, $\Gamma_{k,d}$ were defined in Definition 3.3 and let us explicitly present the scheme obtained in the case where $\mathcal{H}$ is the set of all degree-d monomials, i.e. $\alpha(g^{\mathbf{x}}) = g^{\boldsymbol{\gamma}_{k,d}(\mathbf{x})}$. We denote this scheme by $\mathcal{E}_1$. Theorem 1.2 implies KDM$^{(1)}$-security of $\mathcal{E}_1$ w.r.t. $\hat{\mathcal{F}}_d$, the class of degree-d polynomials in the exponent. In Section 5.3, we show that $\mathcal{E}_1$ actually has stronger security properties.

Encryption scheme $\mathcal{E}_1$. Scheme $\mathcal{E}_1$ is parameterized by $k, d \in \mathbb{N}$ in addition to the parameters of $\mathcal{E}_{\text{BHHO}}[\mathbb{G}, \Gamma_{k,d}]$. We require that $q \cdot 2^{-k} = \text{negl}(\lambda)$, where q is the order of $\mathbb{G}$.

Key generation. On input 1^λ, we generate the secret-key by selecting $\mathbf{x} \xleftarrow{\$} \{0,1\}^k$ and setting $\text{sk} = g^{\mathbf{x}} \in \mathbb{G}^k$. Let $\mathbf{s} = \boldsymbol{\gamma}_{k,d}(\mathbf{x})$, which is uniform in $\Gamma_{k,d}$. We generate the public key according to $\mathcal{E}_{\text{BHHO}}[\mathbb{G}, \Gamma_{k,d}]$, as if $g^{\mathbf{s}}$ was the secret-key. Note that the distribution of public keys is identical to that of $\mathcal{E}_{\text{BHHO}}[\mathbb{G}, \Gamma_{k,d}]$.

Encryption. On inputs pk and w, the encryption algorithm runs the encryption of $\mathcal{E}_{\text{BHHO}}[\mathbb{G}, \Gamma_{k,d}]$ on the same input.

Decryption. On inputs a secret-key $\text{sk} = g^{\mathbf{x}}$ and a ciphertext $\mathbf{c}$, the decryption algorithm first obtains $\mathbf{x}$ from sk, which can be done efficiently since $\mathbf{x} \in \{0,1\}^k$. This enables it, in turn, to compute $\mathbf{s} = \boldsymbol{\gamma}_{k,d}(\mathbf{x})$. Decryption then runs the decryption algorithm of $\mathcal{E}_{\text{BHHO}}[\mathbb{G}, \Gamma_{k,d}]$ with inputs a secret-key $g^{\mathbf{s}}$ and a ciphertext $\mathbf{c}$.

5.3 KDM$^{(n)}$-Security w.r.t. Degree-d Polynomials

We show that the scheme $\mathcal{E}_1$ presented above is in fact KDM$^{(n)}$-secure w.r.t. $\hat{\mathcal{F}}_d$.

Theorem 1.4 (restated)**.** *Scheme $\mathcal{E}_1$, with the parameters described above, is* $\mathrm{KDM}^{(n)}$ *secure w.r.t.* $\hat{\mathcal{F}}_d$*, for any* $n = \mathrm{poly}(\lambda)$*.*

To prove the theorem, we use several additional properties of $\mathcal{E}_{\text{BHHO}}$, stated in Lemma 5.3 below. The proof of the lemma is implicit in [BHHO08] and for the sake of completeness, we also provide a proof in Appendix A.

Lemma 5.3. *Consider $\mathcal{E}_{BHHO}[\mathbb{G}, \mathcal{S}]$ where the* DDH *assumption holds on $\mathbb{G}$. Let $\mathbf{s} \in \mathbb{Z}_q^{\ell}$, $w \in \mathbb{G}$ be arbitrary. Let $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{\ell\times\ell} \times \mathbb{Z}_q^{\ell}$ be an invertible linear transformation on $\mathbb{Z}_q^{\ell}$ and define $\mathbf{s}' = \mathbf{A}\mathbf{s} + \mathbf{b}$.*

Let $\mathrm{pk}, \mathrm{pk}'$ be random variables distributed as public-keys corresponding to $\mathbf{s}, \mathbf{s}'$, respectively. Let $\mathbf{c}, \mathbf{c}'$ be distributed as encryptions of the message w with the public-keys $\mathrm{pk}, \mathrm{pk}'$ respectively. Then the following hold.

- Public-key homomorphism. *There exists an efficiently computable function $P(\mathrm{pk}, \mathbf{A}, \mathbf{b})$ such that the distributions $(\mathrm{pk}, \mathrm{pk}')$ and $(\mathrm{pk}, P(\mathrm{pk}, \mathbf{A}, \mathbf{b}))$ are computationally indistinguishable.*
- Ciphertext homomorphism. *There exists an efficiently computable function $C(\mathbf{c}, \mathbf{A}, \mathbf{b})$ such that the distributions $(\mathrm{pk}, \mathrm{pk}', \mathbf{c}')$ and $(\mathrm{pk}, \mathrm{pk}', C(\mathbf{c}, \mathbf{A}, \mathbf{b}))$ are computationally indistinguishable.*[4]

We can now prove the theorem.

Proof of Theorem 1.4. The proof works by reduction to the $\mathrm{KDM}^{(1)}_{\hat{\mathcal{F}}_d}$-security of $\mathcal{E}_1$ (established in Theorem 1.2). Consider an adversary $\mathcal{A}$ for the $\mathrm{KDM}^{(n)}$ game of $\mathcal{E}_1$ w.r.t. $\hat{\mathcal{F}}_d$. We show that there exists an adversary $\mathcal{B}$ for the $\mathrm{KDM}^{(1)}$ game such that

$$\mathrm{KDM}^{(1)}\mathrm{Adv}[\mathcal{B}, \mathcal{E}_1](\lambda) \geq \mathrm{KDM}^{(n)}\mathrm{Adv}[\mathcal{A}, \mathcal{E}_1](\lambda) - \mathrm{negl}(\lambda) .$$

Initialize. $\mathcal{B}$ gets as input a public key pk that corresponds to some (unknown) secret $\mathbf{x}$. $\mathcal{B}$ samples $\mathbf{y}_1, \ldots, \mathbf{y}_n \xleftarrow{\$} \{0,1\}^k$ and computes $\mathbf{T}_i = \mathsf{T}_{k,d,q}(\mathbf{y}_i)$, where $\mathsf{T}_{k,d,q}$ is taken from Lemma 3.1.

Using the public key homomorphism property, $\mathcal{B}$ generates $\mathrm{pk}_1, \ldots, \mathrm{pk}_n$ where $\mathrm{pk}_i \xleftarrow{\$} P(\mathrm{pk}, \mathbf{T}_i, 0)$ corresponds to the secret $\mathbf{z}_i = \mathbf{x} \oplus \mathbf{y}_i$. $\mathcal{B}$ forwards $\mathrm{pk}_1, \ldots, \mathrm{pk}_n$ to $\mathcal{A}$ as the public keys for the n users.

Queries. $\mathcal{B}$ simulates the query phase of $\mathcal{A}$. Suppose $\mathcal{A}$ makes a query (i, h), where $h \in \hat{\mathcal{F}}_d$. Namely, $h(g^{\mathbf{z}_1}, \ldots, g^{\mathbf{z}_n}) = g^{\varphi(\mathbf{z}_1,\ldots,\mathbf{z}_n)}$, for a degree-$d$ polynomial φ. $\mathcal{B}$ thinks of φ as a polynomial in $\mathbf{x}$ rather than in $\mathbf{z}_1, \ldots, \mathbf{z}_n$. That is, $\mathcal{B}$ computes a degree-d polynomial $\varphi'(\mathbf{x})$ such that $\varphi'(\mathbf{x}) = \varphi(\mathbf{z}_1, \ldots, \mathbf{z}_n)$. This is done by first replacing each variable $z_{i,j}$ in φ with x_i if $y_{i,j} = 0$, or with $1 - x_i$ if $y_{i,j} = 1$; and then computing the coefficients of all the monomials of φ'. This can be done in time $\mathrm{poly}(\ell)$ by opening the parenthesis of φ. Let $h'(g^{\mathbf{x}}) = g^{\varphi'(\mathbf{x})}$.

The next step is sending h' to the challenger and receiving $\mathbf{c}$, an encryption under pk of either either $h'(g^{\mathbf{x}})$ or 0. $\mathcal{B}$ uses the ciphertext homomorphism property to sample $\mathbf{c}' \xleftarrow{\$} C(\mathbf{c}, \mathbf{T}_i, 0)$, which is computationally indistinguishable from an encryption of the same message under pk_i. $\mathcal{B}$ returns $\mathbf{c}'$ to $\mathcal{A}$ as an answer to the query (i, h).

Finish. Upon $\mathcal{A}$'s completion and returning b', $\mathcal{B}$ also terminates and returns the same b'.

We now use a hybrid argument to prove the required claim. For hybrid H_i, let p_i denote the probability that $\mathcal{A}$ returns $b' = b$.

[4] Note that $C(\cdot)$ does not take pk' as input. Therefore, this property also implies that two independent public-keys that correspond to the same secret-key generate two computationally indistinguishable ciphertext distributions.

1. In hybrid H_0, $\mathcal{A}$ interacts with the simulator $\mathcal{B}$ as described above. By definition,
$$\mathrm{KDM}^{(1)}\mathrm{Adv}[\mathcal{B}, \mathcal{E}_1](\lambda) = |p_0 - 1/2| \ .$$
2. In hybrid H_1, $\mathcal{A}$ interacts with with a simulator identical to $\mathcal{B}$ with one change: rather than sample from the distribution $P(\mathrm{pk}, \mathbf{T}_i, 0)$, in H_1 the simulator samples an actual public-key for $\mathbf{z}_i$. Lemma 5.3 implies that $|p_1 - p_0| = \mathrm{negl}(\lambda)$, since otherwise we can consider a hybrid $H_1^{(j)}$ where the first j keys are produced according to $P(\mathrm{pk}, \mathbf{T}_i, 0)$ and the rest are properly generated. Two adjacent hybrids which are computationally distinguishable enable to find $\mathbf{s}, \mathbf{A}, \mathbf{b}$ that contradict public-key homomorphism.
3. In hybrid H_2, $\mathcal{A}$ interacts with a simulator identical to that of H_1, with one change: rather than sampling from the distribution $C(\mathbf{c}, \mathbf{T}_i, 0)$, in H_2 the simulator samples an actual encryption of the relevant message with public-key pk_i. Again, $|p_2 - p_1| = \mathrm{negl}(\lambda)$ since otherwise we can define $H_2^{(j)}$ where the first j encryptions are obtained using ciphertext homomorphism and the rest are properly generated. This, in turn, will imply a distinguisher for ciphertext homomorphism.

   Noting that, H_2 is identical to the $\mathrm{KDM}^{(n)}$-game of $\mathcal{A}$, we get that $\mathrm{KDM}^{(n)}\mathrm{Adv}[\mathcal{A}, \mathcal{E}_1](\lambda) = |p_2 - 1/2|$.

We conclude that $\mathrm{KDM}^{(1)}\mathrm{Adv}[\mathcal{B}, \mathcal{E}_1](\lambda) \geq \mathrm{KDM}^{(n)}\mathrm{Adv}[\mathcal{A}, \mathcal{E}_1](\lambda) - \mathrm{negl}(\lambda)$ as required.[5] □

6 LWE Based KDM Security

In this section we show similar results to those of Section 5, this time under the LWE assumption. We follow the same general outline. First, in Section 6.1, we present the relevant previous work, in this case - the scheme of [ACPS09], denoted $\mathcal{E}_{\mathrm{ACPS}}$. Then, in Section 6.2, we prove the entropy-k $\mathrm{KDM}^{(1)}$-security of $\mathcal{E}_{\mathrm{ACPS}}$ w.r.t. affine functions $\mathcal{F}_{\mathrm{aff}}$, and present the consequences of applying Theorem 1.1 to $\mathcal{E}_{\mathrm{ACPS}}$. Finally, in Section 6.3, we show that in the special case of degree-d polynomials, we can in fact prove $\mathrm{KDM}^{(n)}$-security of the scheme obtained from Theorem 1.1.

Preliminaries. In this section, we use distributions that are derived from Gaussians. For any $\sigma > 0$, we denote $D_\sigma(x) = e^{-\pi(x/\sigma)^2}/\sigma$, the (scaled) density function of the one dimensional Gaussian distribution. For any $q \in \mathbb{N}$ and $\sigma > 0$ we define $\bar{\Psi}_\sigma$ to be the distribution over $\mathbb{Z}_q$ obtained by sampling $y \xleftarrow{\$} D_\sigma$ and outputting $\lfloor q \cdot y \rceil \pmod q$. We define $D_{\mathbb{Z}^m,\sigma}$ to be the distribution over all $\mathbf{x} \in \mathbb{Z}^m$ such that $\Pr[\mathbf{x}]$ is proportional to $\prod_{i \in [m]} D_\sigma(x_i)$. We note that this distribution is efficiently sampleable for any $\sigma > 0$.

6.1 Scheme $\mathcal{E}_{\mathrm{ACPS}}$

We present the $\mathcal{E}_{\mathrm{ACPS}}[\mathcal{S}]$ scheme which is similar to the scheme presented in [ACPS09]. The only difference is that we take the distribution of secret-keys as a parameter. We also use slightly different notation for consistency with the rest of this paper.

[5] In our proof, the number of hybrids $H_1^{(j)}$ and $H_2^{(j)}$ depend on n and on the number of queries made by $\mathcal{A}$ (respectively). These parameters, therefore, factor into the advantage of the DDH adversary obtained in the reduction. We remark that using a more complicated version of Lemma 5.3, it is possible to achieve a more efficient reduction where the number of hybrids is $O(\ell)$, regardless of n and $\mathcal{A}$ (as in the security proof of [BHHO08]). For our purposes, however, the simpler version suffices.

Parameters. Let p be a prime and $q = p^2$. We set $\ell, m \in \mathbb{N}$ to be polynomial functions of λ such that $m \geq 2(\ell+1)\log q$. Let $\chi = \bar{\Psi}_\sigma$ for $\sigma = \sigma(\lambda) \in (0,1)$ such that $\sigma \leq \frac{1}{p \cdot \sqrt{m} \cdot \omega(\log \lambda)}$. We also fix some $\tau = \omega(\sqrt{\log \lambda})$. Finally, let $\mathcal{S} \subseteq \mathbb{Z}_p^\ell$. The secret-key space is $\mathcal{S}$ and the message space is $\mathbb{Z}_p$.

Key generation. On input 1^λ, sample $\mathbf{s} \xleftarrow{\$} \mathcal{S}$ and set $\mathrm{sk} = \mathbf{s}$.[6] Then, sample $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m \times \ell}$ and $\boldsymbol{\eta} \xleftarrow{\$} \chi^m$ and set $\mathrm{pk} = (\mathbf{A}, \mathbf{A} \cdot \mathbf{s} + \boldsymbol{\eta}) \in \mathbb{Z}_q^{m \times \ell} \times \mathbb{Z}_q^m$.

Encryption. Define the distribution $E_{\mathbf{A},\mathbf{b}}$ in $\mathbb{Z}_q^\ell \times \mathbb{Z}_q$ as follows. $E_{\mathbf{A},\mathbf{b}}$ samples $\mathbf{r} \xleftarrow{\$} D_{\mathbb{Z}^m, \tau}$, $e \xleftarrow{\$} \bar{\Psi}_{\tau'}$ where $\tau' = \tau\sqrt{m}(\sigma + \frac{1}{2q})$ and outputs $(\mathbf{A}^T \cdot \mathbf{r}, \mathbf{b}^T \cdot \mathbf{r} + e) \in \mathbb{Z}_q^\ell \times \mathbb{Z}_q$.

On input a public key $\mathrm{pk} = (\mathbf{A}, \mathbf{b})$ and a message $w \in \mathbb{Z}_p$, the encryption algorithm samples $(\mathbf{u}, v) \xleftarrow{\$} E_{\mathbf{A},\mathbf{b}}$ and outputs

$$(\mathbf{u}, v + w \cdot p) \ .$$

Decryption. On input a secret key $\mathbf{s}$ and a ciphertext $(\mathbf{u}, c)$, the decryption algorithm outputs

$$\left\lfloor \left(c - \mathbf{u}^T \cdot \mathbf{s} \pmod q\right)/p \right\rceil \pmod p \ .$$

The proof of correctness provided in [ACPS09] applies to any $\mathbf{s} \in \mathbb{Z}_p^\ell$. It states that correctness holds if $\sigma \leq \frac{1}{p \cdot \sqrt{m} \cdot \omega(\log \lambda)}$. In addition, they provide a few lemmas that we will use in the remainder of this section. Let us state them here.

The first lemma shows that given a public key, it is possible (with all but negligible probability) to generate encryptions of affine functions of the secret key $\mathbf{s}$ without knowing $\mathbf{s}$. This is useful for simulating the KDM game without knowing the secret key.

Lemma 6.1 ([ACPS09, Lemma 5])**.** *For all $\mathbf{s} \in \mathbb{Z}_p^\ell$, $(\mathbf{t}, w) \in \mathbb{Z}_p^\ell \times \mathbb{Z}_p$, with all but negligible probability over $\mathbf{A}, \boldsymbol{\eta}$ it holds that for $(\mathbf{u}, v) \xleftarrow{\$} E_{(\mathbf{A},\mathbf{b})}$,*

$$\mathsf{SD}((\mathbf{u}, v + (\mathbf{t}^T \cdot \mathbf{s} + w) \cdot p), ((\mathbf{u}, v) + (-\mathbf{t} \cdot p, w \cdot p))) = \mathrm{negl}(\lambda) \ .$$

Note that $(\mathbf{u}, v + (\mathbf{t}^T \cdot \mathbf{s} + w) \cdot p)$ is the distribution of encryptions of $\mathbf{t}^T \cdot \mathbf{s} + w$ under public-key $(\mathbf{A}, \mathbf{b})$.

The second lemma shows that if the $\mathbf{b}$ component of the public key is sampled uniformly (i.e. independently of $\mathbf{s}$), then the resulting encryption scheme almost always generates uniformly distributed ciphertexts. This is useful since the real distribution of $\mathbf{b}$ is computationally indistinguishable from uniform, which enables us to claim that "real" public keys generate ciphertexts which are computationally indistinguishable from uniform.

Lemma 6.2 ([ACPS09, Lemma 6])**.** *With all but negligible probability over $(\mathbf{A}, \mathbf{b}) \xleftarrow{\$} \mathbb{Z}_q^{m \times \ell} \times \mathbb{Z}_q^m$ it holds that $\mathsf{SD}(E_{(\mathbf{A},\mathbf{b})}, U(\mathbb{Z}_q^\ell \times \mathbb{Z}_q)) = \mathrm{negl}(\lambda)$.*

6.2 Amplification of $\mathrm{KDM}^{(1)}$ Security

We state and prove a theorem analogous to Theorem 1.2. Recall that a class of functions $\mathcal{H} = \{h_1, \ldots, h_\ell\}$ over the same domain is entropy preserving if the function $\alpha_{\mathcal{H}}(x) = (h_1(x), \cdots, h_\ell(x))$ is injective.

[6] In [ACPS09], $\mathbf{s}$ is sampled from the distribution χ^ℓ.

Theorem 1.3 (restated)**.** *Let p be a prime number that is super-polynomial in λ and denote $q = p^2$. Let m, ℓ, σ, χ be as in the parameters of $\mathcal{E}_{ACPS}$. Let $k \leq \ell$ and set $k' = \frac{k-\omega(\log\lambda)}{\log q}$. Let $\beta = \beta(\lambda) \in (0,1)$ be such that $\frac{\beta}{\sigma} = \mathrm{negl}(\lambda)$ and denote $\chi' = \bar{\Psi}_\beta$. Let $\mathcal{H} = \{h_1, \ldots, h_\ell : h_i \in \{0,1\}^k \to \{0,1\}\}$ be an entropy preserving class of efficiently computable functions with cardinality $\ell = \mathrm{poly}(\lambda)$. Then under the $\mathrm{LWE}_{q,m,k',\chi'}$ assumption, there exists a public-key encryption scheme that is $\mathrm{KDM}^{(1)}$ secure w.r.t. function class*

$$\mathcal{F}_\mathcal{H} = \left\{ f(\mathbf{x}) = \sum_{i\in[\ell]} t_i h_i(\mathbf{x}) + w \pmod p : (\mathbf{t}, w) \in \mathbb{Z}_p^\ell \times \mathbb{Z}_p \right\} .$$

Before giving the proof, let us discuss the parameters of the assumption we rely on. The decisional $\mathrm{LWE}_{q,m,k',\chi'}$ assumption (see Section 2.1) is equivalent to the search version under a $\mathrm{poly}(q)$-time reduction. The search version, in turn, is shown in [Reg05] to correspond to worst-case lattice problems, under quantum reductions. In [Pei09], a classical reduction from other worst-case lattice problems to search LWE is shown. Thus, we can set p and q to be quasi-polynomial in λ, set $\beta \geq n/q$ and set $\frac{\sigma}{\beta}$ to be quasi-polynomial in λ as well (recall that for correctness we must take $\sigma \leq \frac{1}{p\cdot\sqrt{m}\cdot\omega(\log\lambda)}$, so we cannot set σ to be too large, but one can verify that a proper selection of parameters does exist). Using such parameters we can relate the security of our scheme to either the worst case hardness of obtaining a quasi-polynomial approximation factor for a lattice problem such as GapSVP, using quasi-polynomial time quantum algorithms, or to the worst case hardness of obtaining a classical quasi-polynomial time algorithm for a lattice problem such as $\mathrm{GapSVP}_{\zeta,\gamma}$ with quasi-polynomial ζ.

To prove Theorem 1.3, we employ Theorem 1.1. As a precondition, we will need to establish entropy-k $\mathrm{KDM}^{(1)}$-security for $\mathcal{E}_{\mathrm{ACPS}}$. Unlike in the case of the DDH scheme $\mathcal{E}_{\mathrm{BHHO}}$, this is not straightforward. We do this in two steps. First, we prove $\mathrm{KDM}^{(1)}$-security based on a nonstandard assumption (see Definition 6.1 below). Then, we use a novel result of Goldwasser, Kalai, Peikert and Vaikuntanathan [GKPV09] that implies that for the parameters of Theorem 1.3, LWE reduces to our new assumption, thus ultimately basing our scheme on standard decisional LWE. We remark that it may be possible to achieve better parameters than those stated in Theorem 1.3 using a more efficient reduction, if such exists.

We proceed by presenting the new assumption. Intuitively, recalling that the key generation of $\mathcal{E}_{\mathrm{ACPS}}$ is just generating an LWE instance, our new assumption is that LWE holds even if the secret-key $\mathbf{s}$ only has min-entropy k rather than being uniformly sampled.

Definition 6.1 (entropy LWE assumptions)**.** *Consider the distributions $(\mathbf{A}, \mathbf{As}+\mathbf{x})$ and $(\mathbf{A}, \mathbf{u})$ in the $\mathrm{LWE}_{q,m,\ell,\chi}$ assumption, with the only difference being that $\mathbf{s} \xleftarrow{\$} \mathcal{S}$, for some set $\mathcal{S} \subseteq \mathbb{Z}_q^\ell$ (instead of $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^\ell$).* The $\mathrm{LWE}_{q,m,\ell,\chi}[\mathcal{S}]$ assumption *is that these distributions are computationally indistinguishable.* The entropy-k $\mathrm{LWE}_{q,m,\ell,\chi}$ assumption *is that the $\mathrm{LWE}_{q,m,\ell,\chi}[\mathcal{S}]$ assumption holds for all $\mathcal{S} \subseteq \{0,1\}^\ell$ with $|\mathcal{S}| \geq 2^k$.*

The following lemma establishes the entropy-k $\mathrm{KDM}^{(1)}$-security of $\mathcal{E}_{\mathrm{ACPS}}[\mathcal{S}]$, based on the $\mathrm{LWE}_{q,m,\ell,\chi}[\mathcal{S}]$ assumption. The proof is similar in spirit to [ACPS09, Theorem 2] and is deferred to Appendix B. Recall that $\mathcal{F}_{\mathrm{aff}} = \{f_{\mathbf{t},w}(\mathbf{x}) = \mathbf{t}^T\mathbf{x} + w : (\mathbf{t}, w) \in \mathbb{Z}_p^k \times \mathbb{Z}_p\}$ is the set of affine functions over $\mathbb{Z}_p$.

Lemma 6.3. *Let $\mathcal{S} \subseteq \mathbb{Z}_p^\ell$. If $\mathrm{LWE}_{q,m,\ell,\chi}[\mathcal{S}]$ holds, then $\mathcal{E}_{ACPS}[\mathcal{S}]$ is $\mathrm{KDM}^{(1)}$ secure w.r.t. $\mathcal{F}_{\mathit{aff}}$.*

In a recent work, standard decisional LWE is reduced to entropy-k LWE.

Theorem 6.4 ([GKPV09, Theorem 1]). *Let p be a prime of super-polynomial size, and set $q = p^e$ for some constant e. Let $m, \ell = \mathrm{poly}(\lambda)$. Let $k \le \ell$ and set $k' = \frac{k-\omega(\log \lambda)}{\log q}$. Let $\sigma, \beta \in (0,1)$ such that $\beta/\sigma = \mathrm{negl}(\lambda)$ and set $\chi = \bar{\Psi}_\sigma$, $\chi' = \bar{\Psi}_\beta$. Then if the $\mathrm{LWE}_{q,m,k',\chi'}$ assumption holds then the entropy-k $\mathrm{LWE}_{q,m,\ell,\chi}$ assumption holds as well.*

We remark that [GKPV09] only prove this for $e = 1$ (i.e. prime q) but the same proof can be used for any constant (specifically for $e = 2$ which is used here).

The proof of Theorem 1.3 now follows.

Proof of Theorem 1.3. Fix a function class $\mathcal{H}$ as in the theorem statement. By Lemma 6.3, it holds that under the entropy-k $\mathrm{LWE}_{q,m,\ell,\chi}$ assumption, $\mathcal{E}_{\mathrm{ACPS}}[\{0,1\}^\ell]$ is entropy-k $\mathrm{KDM}^{(1)}$-secure w.r.t. $\mathcal{F}_{\mathrm{aff}}$. Thus we can apply Theorem 1.1, setting $\alpha(\mathbf{x}) = (h_1(\mathbf{x}), \cdots, h_\ell(\mathbf{x}))$, and obtain a $\mathrm{KDM}^{(1)}$-secure scheme w.r.t. $\mathcal{F}_{\mathcal{H}}$, under the entropy-$k$ $\mathrm{LWE}_{q,m,\ell,\chi}$ assumption. To finish the proof, we use Theorem 6.4 to argue that the $\mathrm{LWE}_{q,m,k',\chi'}$ assumption implies the entropy-k $\mathrm{LWE}_{q,m,\ell,\chi}$ assumption. □

In the specific case of using the set of all degree-d monomials as the function class $\mathcal{H}$, we obtain a $\mathrm{KDM}^{(1)}$-secure scheme w.r.t. $\mathcal{F}_d$, all degree-d polynomials modulo p. We describe this scheme, $\mathcal{E}_2$, explicitly. In Section 6.3 we show that $\mathcal{E}_2$ is in fact $\mathrm{KDM}^{(n)}$ secure w.r.t. $\mathcal{F}_d$. Recall that $\boldsymbol{\gamma}_{k,d}$, $\nu_{k,d}$, $\Gamma_{k,d}$ were defined in Definition 3.3

Encryption scheme $\mathcal{E}_2$. Let $k, d \in \mathbb{N}$ and consider $p, q, m, \sigma, \chi, \tau$, as in the definition of $\mathcal{E}_{\mathrm{ACPS}}[\Gamma_{k,d}]$, specifically let $\ell = \nu_{k,d}$. The secret-key space of $\mathcal{E}_2$ is $\{0,1\}^k$ and the message space is $\mathbb{Z}_p$.

Key generation. On input 1^λ, select $\mathbf{x} \stackrel{\$}{\leftarrow} \{0,1\}^k$ and set $\mathrm{sk} = \mathbf{x}$. We denote $\mathbf{s} = \boldsymbol{\gamma}_{k,d}(\mathbf{x})$ and note that $\mathbf{s}$ is uniform in $\Gamma_{k,d}$. The public key pk is generated as in $\mathcal{E}_{\mathrm{ACPS}}[\Gamma_{k,d}]$. Namely, $\mathrm{pk} = (\mathbf{A}, \mathbf{A} \cdot \mathbf{s} + \boldsymbol{\eta}) \in \mathbb{Z}_q^{m \times \ell} \times \mathbb{Z}_q^m$. Note that the distributions of the public keys in $\mathcal{E}_2$ and $\mathcal{E}_{\mathrm{ACPS}}[\Gamma_{k,d}]$ are identical.

Encryption. On inputs a public-key pk and message w, the encryption algorithm runs the encryption algorithm of $\mathcal{E}_{\mathrm{ACPS}}[\Gamma_{k,d}]$ with the same inputs.

Decryption. On inputs a secret-key $\mathrm{sk} = \mathbf{x} \in \{0,1\}^k$ and a ciphertext $(\mathbf{u}, c)$, the decryption algorithm uses $\mathbf{x}$ to obtain $\mathbf{s} = \boldsymbol{\gamma}_{k,d}(\mathbf{x})$. Decryption then proceeds as in $\mathcal{E}_{\mathrm{ACPS}}[\Gamma_{k,d}]$, with inputs a secret-key $\mathbf{s}$ and a ciphertext $(\mathbf{u}, c)$.

6.3 $\mathrm{KDM}^{(n)}$-Security w.r.t. Degree-d Polynomials

We show that $\mathcal{E}_2$ is $\mathrm{KDM}^{(n)}$-secure w.r.t. $\mathcal{F}_d$.

Theorem 1.5 (restated). *Consider the scheme $\mathcal{E}_2$ with p being super-polynomial in λ. Let $k' = \frac{k-\omega(\log \lambda)}{\log q}$ and let $\beta = \beta(\lambda) \in (0,1)$ be such that $\frac{\beta}{\sigma} = \mathrm{negl}(\lambda)$. Define $\chi' = \bar{\Psi}_\beta$. Under the $\mathrm{LWE}_{q,m\cdot n,k',\chi'}$ assumption, $\mathcal{E}_2$ is $\mathrm{KDM}^{(n)}$-secure w.r.t. the class of degree-d polynomials modulo p.*

Note that if $\mathrm{LWE}_{q,m\cdot n,k',\chi'}$ is hard for all $n = \mathrm{poly}(\lambda)$, then $\mathcal{E}_2$ is $\mathrm{KDM}^{(n)}$-secure for any polynomial number of "users". We also note that as in Theorem 1.3, the LWE assumption we rely on is related to worst-case lattice problems. See discussion in Section 6.2 for more details.

Proof. By Theorem 6.4, the $\mathrm{LWE}_{q,m\cdot n,k',\chi'}$ assumption implies the entropy-k $\mathrm{LWE}_{q,m\cdot n,\ell,\chi}$ assumption which, in turn, implies the $\mathrm{LWE}_{q,m\cdot n,\ell,\chi}[\Gamma_{k,d}]$ assumption. Therefore, it suffices to prove the $\mathrm{KDM}^{(n)}$-security of $\mathcal{E}_2$ based on the $\mathrm{LWE}_{q,m\cdot n,\ell,\chi}[\Gamma_{k,d}]$ assumption.

Let $\mathcal{A}$ be an adversary for the $\mathrm{KDM}^{(n)}_{\mathcal{F}_d}$ game of $\mathcal{E}_2$. We present an adversary $\mathcal{B}$ such that

$$\mathrm{LWE}_{q,(m\cdot n),\ell,\chi}[\Gamma_{k,d}]\mathrm{Adv}[\mathcal{B}](\lambda) \geq \mathrm{KDM}^{(n)}_{\mathcal{F}_d}\mathrm{Adv}[\mathcal{A},\mathcal{E}_2](\lambda) - \mathrm{negl}(\lambda)\ .^{7}$$

The input to $\mathcal{B}$ is $(\mathbf{A},\mathbf{b}) \in (\mathbb{Z}_q^{(mn)\times\ell} \times \mathbb{Z}_q^{mn})$. We represent them as a sequence of n pairs $(\mathbf{A}_i,\mathbf{b}_i) \in (\mathbb{Z}_q^{m\times\ell} \times \mathbb{Z}_q^{m})$ where $\mathbf{A}_i$ is uniform and $\mathbf{b}_i$ is either $\mathbf{b}_i^{(0)} = \mathbf{A}_i\mathbf{s}+\boldsymbol{\eta}_i$ for $\mathbf{s} \stackrel{\$}{\leftarrow} \Gamma_{k,d}$, $\boldsymbol{\eta}_i \stackrel{\$}{\leftarrow} \chi^m$, or $\mathbf{b}_i^{(1)} \stackrel{\$}{\leftarrow} \mathbb{Z}_q^m$. Let $\mathbf{x}$ be such that $\boldsymbol{\gamma}_{k,d}(\mathbf{x}) = \mathbf{s}$.

$\mathcal{B}$ simulates the $\mathrm{KDM}^{(n)}_{\mathcal{F}_d}$ game for $\mathcal{A}$.

Initialize. $\mathcal{B}$ flips a coin $\xi \stackrel{\$}{\leftarrow} \{0,1\}$. It also selects $\mathbf{y}_i \stackrel{\$}{\leftarrow} \{0,1\}^k$ for all $i \in [n]$ and computes $\mathbf{T}_i = \mathsf{T}_{k,d,q}(\mathbf{y}_i) \in \mathbb{Z}_q^{\ell\times\ell}$, where $\mathsf{T}_{k,d,q}$ is defined in Lemma 3.1. Denote $\mathbf{z}_i = \mathbf{x} \oplus \mathbf{y}_i$, $\mathbf{C}_i = \mathbf{A}_i\mathbf{T}_i$. Recall that for all $\mathbf{x} \in \{0,1\}^k$ and $i \in [n]$ it holds that $\mathbf{T}_i \cdot \boldsymbol{\gamma}_{k,d}(\mathbf{x}) = \boldsymbol{\gamma}_{k,d}(\mathbf{x} \oplus \mathbf{y}_i) = \boldsymbol{\gamma}_{k,d}(\mathbf{z}_i)$ and $\mathbf{T}_i^2 \cdot \boldsymbol{\gamma}_{k,d}(\mathbf{x}) = \boldsymbol{\gamma}_{k,d}(\mathbf{x})$. Notice that $\mathbf{A}_i \cdot \mathbf{s} = \mathbf{A}_i \cdot \mathbf{T}_i^2 \cdot \boldsymbol{\gamma}_{k,d}(\mathbf{x}) = \mathbf{C}_i \cdot \boldsymbol{\gamma}_{k,d}(\mathbf{z}_i)$. This, together with the fact that $\mathbf{C}_i$ is uniformly distributed,[8] implies that $\{(\mathbf{z}_i, (\mathbf{C}_i, \mathbf{b}_i^{(0)}))\}_{i\in[n]}$ is a legally distributed set of n secret and public keys for $\mathcal{E}_2$. $\mathcal{B}$ sets $\mathrm{pk}_i = (\mathbf{C}_i, \mathbf{b}_i)$ and sends $\mathrm{pk}_1, \ldots, \mathrm{pk}_n$ to $\mathcal{A}$.

Queries. When $\mathcal{A}$ makes a query (j,φ) where $\varphi(\mathbf{z}_1,\ldots,\mathbf{z}_n)$ is a degree-d polynomial in all secret keys, $\mathcal{B}$ uses the vectors $\mathbf{y}_1,\ldots,\mathbf{y}_n$ to find a degree-d polynomial φ' such that $\varphi'(\mathbf{z}_j) = \varphi(\mathbf{z}_1,\ldots,\mathbf{z}_n)$. This is possible since

$$\varphi(\mathbf{z}_1,\ldots,\mathbf{z}_n) = \varphi(\mathbf{x}\oplus\mathbf{y}_1,\ldots,\mathbf{x}\oplus\mathbf{y}_n) = \varphi(\mathbf{z}_j \oplus (\mathbf{y}_j \oplus \mathbf{y}_1),\ldots,\mathbf{z}_j \oplus (\mathbf{y}_j \oplus \mathbf{y}_n))$$

which means we can replace each variable $z_{i,i'}$ in φ with either $z_{j,i'}$ if $(\mathbf{y}_j \oplus \mathbf{y}_i)_{i'} = 0$ or with $1 - z_{j,i'}$ if $(\mathbf{y}_j \oplus \mathbf{y}_i)_{i'} = 1$. Opening the parenthesis and computing the coefficients of all the monomials (which can be done in time $\mathrm{poly}(\ell)$) produces the required φ', or in other words, the coefficients vector $\mathbf{t} \in \mathbb{Z}_p^\ell$ such that $\varphi'(\mathbf{z}_j) = \mathbf{t}^T\boldsymbol{\gamma}_{k,d}(\mathbf{z}_j)$ (recall that $\boldsymbol{\gamma}_{k,d}(\cdot)$ contains the free coefficient and thus we do not need to add it explicitly).

Then, $\mathcal{B}$ samples $(\mathbf{u}, v) \stackrel{\$}{\leftarrow} E_{(\mathbf{C}_j,\mathbf{b}_j)}$ and sets $\mathbf{c}_0 = (\mathbf{u}, v) + (-\mathbf{t}\cdot p, 0)$ and $\mathbf{c}_1 = (\mathbf{u}, v)$. $\mathcal{B}$ then returns $\mathbf{c}_\xi$ as an answer to $\mathcal{A}$.

Finish. When $\mathcal{A}$ terminates and returns ξ', $\mathcal{B}$ returns 1 if $\xi' = \xi$ and 0 otherwise.

The analysis is almost identical to that of Lemma 6.3: if $\mathbf{b}_i = \mathbf{b}_i^{(0)}$, then $(\mathbf{C}_i, \mathbf{b}_i)$ is a legal public-key for $\mathcal{E}_2$, that corresponds to secret key $\mathbf{z}_i$. In this case, by Lemma 6.1, $\mathcal{B}$ simulates the $\mathrm{KDM}^{(n)}_{\mathcal{F}_d}$ game up to a negligible statistical distance, and thus $\left|\Pr[\mathcal{B}(\mathbf{A},\mathbf{b}^{(0)}) = 1] - \Pr[\mathcal{A} \text{ wins } \mathrm{KDM}^{(n)}]\right| = \mathrm{negl}(\lambda)$. However, if $\mathbf{b}_i = \mathbf{b}_i^{(1)}$ then by Lemma 6.2, $\mathbf{c}_0, \mathbf{c}_1$ are within negligible statistical distance and thus the views of $\mathcal{A}$ where $\xi = 0$ and where $\xi = 1$ are within negligible statistical distance. Therefore, $\left|\Pr[\mathcal{B}(\mathbf{A},\mathbf{b}^{(1)}) = 1] - \frac{1}{2}\right| = \mathrm{negl}(\lambda)$, and we conclude that

$$\left|\Pr[\mathcal{B}(\mathbf{A},\mathbf{b}^{(0)}) = 1] - \Pr[\mathcal{B}(\mathbf{A},\mathbf{b}^{(1)}) = 1]\right| \geq \left|\Pr[\mathcal{A} \text{ wins } \mathrm{KDM}^{(n)}] - \frac{1}{2}\right| - \mathrm{negl}(\lambda)$$

as required. □

[7] Unlike Theorem 1.4, the reduction here is directly to the cryptographic assumption. This is done to achieve better parameters.

[8] We remark that this is not straightforward since $\mathbb{Z}_q$ is not a field, however it is true in our case.

References

[ABHS05] 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, *ESORICS*, volume 3679 of *Lecture Notes in Computer Science*, pages 374–396. Springer, 2005.

[ACPS09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In Halevi [Hal09], pages 595–618.

[AGV09] Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In Reingold [Rei09], pages 474–495.

[BDU08] Michael Backes, Markus Dürmuth, and Dominique Unruh. Oaep is secure under key-dependent messages. In Josef Pieprzyk, editor, *ASIACRYPT*, volume 5350 of *Lecture Notes in Computer Science*, pages 506–523. Springer, 2008.

[BHHO08] Dan Boneh, Shai Halevi, Michael Hamburg, and Rafail Ostrovsky. Circular-secure encryption from decision diffie-hellman. In David Wagner, editor, *CRYPTO*, volume 5157 of *Lecture Notes in Computer Science*, pages 108–125. Springer, 2008.

[BN00] Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In Okamoto [Oka00], pages 531–545.

[BPS07] Michael Backes, Birgit Pfitzmann, and Andre Scedrov. Key-dependent message security under active attacks - brsim/uc-soundness of symbolic encryption with key cycles. In *CSF*, pages 112–124. IEEE Computer Society, 2007.

[BR00] Mihir Bellare and Phillip Rogaway. Encode-then-encipher encryption: How to exploit nonces or redundancy in plaintexts for efficient cryptography. In Okamoto [Oka00], pages 317–330.

[BRS02] 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*, volume 2595 of *Lecture Notes in Computer Science*, pages 62–75. Springer, 2002.

[CCS09] Jan Camenisch, Nishanth Chandran, and Victor Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In Antoine Joux, editor, *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 351–368. Springer, 2009.

[CL01] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In Birgit Pfitzmann, editor, *EUROCRYPT*, volume 2045 of *Lecture Notes in Computer Science*, pages 93–118. Springer, 2001.

[DDN00] Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM J. Comput.*, 30(2):391–437, 2000.

[Gam84] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *CRYPTO*, pages 10–18, 1984.

[GKPV09] Shafi Goldwasser, Yael Kalai, Chris Peikert, and Vinod Vaikuntanathan. Robustness of the learning with errors assumption. Manuscript, 2009.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[Gol04] Oded Goldreich. *Foundations of Cryptography - Basic Applications*. Cambridge University Press, 2004.

[Hal09] Shai Halevi, editor. *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, volume 5677 of *Lecture Notes in Computer Science*. Springer, 2009.

[HH09] Iftach Haitner and Thomas Holenstein. On the (im)possibility of key dependent encryption. In Reingold [Rei09], pages 202–219.

[HK07] Shai Halevi and Hugo Krawczyk. Security under key-dependent inputs. In Peng Ning, Sabrina De Capitani di Vimercati, and Paul F. Syverson, editors, *ACM Conference on Computer and Communications Security*, pages 466–475. ACM, 2007.

[HU08] Dennis Hofheinz and Dominique Unruh. Towards key-dependent message security in the standard model. In Nigel P. Smart, editor, *EUROCRYPT*, volume 4965 of *Lecture Notes in Computer Science*, pages 108–126. Springer, 2008.

[Kra01] Hugo Krawczyk. The order of encryption and authentication for protecting communications (or: How secure is ssl?). In Joe Kilian, editor, *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 310–331. Springer, 2001.

[KY00] Jonathan Katz and Moti Yung. Unforgeable encryption and chosen ciphertext secure modes of operation. In Bruce Schneier, editor, *FSE*, volume 1978 of *Lecture Notes in Computer Science*, pages 284–299. Springer, 2000.

[LC03] Peeter Laud and Ricardo Corin. Sound computational interpretation of formal encryption with composed keys. In Jong In Lim and Dong Hoon Lee, editors, *ICISC*, volume 2971 of *Lecture Notes in Computer Science*, pages 55–66. Springer, 2003.

[Mit09] Michael Mitzenmacher, editor. *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*. ACM, 2009.

[NS09] Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In Halevi [Hal09], pages 18–35.

[Oka00] Tatsuaki Okamoto, editor. *Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings*, volume 1976 of *Lecture Notes in Computer Science*. Springer, 2000.

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In Mitzenmacher [Mit09], pages 333–342.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *STOC*, pages 84–93. ACM, 2005.

[Rei09] Omer Reingold, editor. *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009, San Francisco, CA, USA, March 15-17, 2009. Proceedings*, volume 5444 of *Lecture Notes in Computer Science*. Springer, 2009.

[RS91] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Joan Feigenbaum, editor, *CRYPTO*, volume 576 of *Lecture Notes in Computer Science*, pages 433–444. Springer, 1991.

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

A Proof of Lemma 5.3

We use the fact that under the DDH assumption, the distributions $(g^{\mathbf{z}}, g^{\mathbf{y}})$ and $(g^{\mathbf{z}}, g^{r\mathbf{z}})$, where $\mathbf{z}, \mathbf{y} \xleftarrow{\$} \mathbb{Z}_q^\ell$, $r \xleftarrow{\$} \mathbb{Z}_q$, are computationally indistinguishable. This is true by definition for $\ell = 2$ and extends easily for any polynomial ℓ.

Public-key homomorphism. The function $P(\mathrm{pk}, \mathbf{A}, \mathbf{b})$ is defined as follows. For $\mathrm{pk} = (g^{\mathbf{z}}, g^v)$, it samples $r \xleftarrow{\$} \mathbb{Z}_q$ and outputs $(g^{\mathbf{A}^{-T}\mathbf{z}\cdot r}, g^{v\cdot r} \cdot g^{-r\cdot\mathbf{z}^T\mathbf{A}^{-1}\mathbf{b}})$. It remains to prove that

$$(\mathrm{pk}, \mathrm{pk}') = ((g^{\mathbf{z}}, g^{-\mathbf{z}^T\mathbf{s}}), (g^{\mathbf{y}}, g^{-\mathbf{y}^T(\mathbf{A}\mathbf{s}+\mathbf{b})}))$$

is computationally indistinguishable from

$$(\mathrm{pk}, P(\mathrm{pk}, \mathbf{A}, \mathbf{b})) = ((g^{\mathbf{z}}, g^{-\mathbf{z}^T\mathbf{s}}), (g^{\mathbf{A}^{-T}\mathbf{z}\cdot r}, g^{-\mathbf{z}^T\mathbf{s}\cdot r} \cdot g^{-r\cdot\mathbf{z}^T\mathbf{A}^{-1}\mathbf{b}})) ,$$

for $\mathbf{z}, \mathbf{y} \xleftarrow{\$} \mathbb{Z}_q^\ell$, $r \xleftarrow{\$} \mathbb{Z}_q$. To do this, we denote $\mathbf{y}' = \mathbf{A}^{-T}\mathbf{z} \cdot r$ and notice that

$$(\mathrm{pk}, P(\mathrm{pk}, \mathbf{A}, \mathbf{b})) = ((g^{\mathbf{z}}, g^{-\mathbf{z}^T\mathbf{s}}), (g^{\mathbf{y}'}, g^{-\mathbf{y}'^T(\mathbf{A}\mathbf{s}+\mathbf{b})})) .$$

Therefore, it is sufficient to prove that $(g^{\mathbf{z}}, g^{\mathbf{y}})$ is computationally indistinguishable from $(g^{\mathbf{z}}, g^{\mathbf{y}'})$. Since, as we mentioned, $(g^{\mathbf{z}}, g^{\mathbf{y}})$ and $(g^{\mathbf{z}}, g^{r\mathbf{z}})$ are computationally indistinguishable, and since $\mathbf{A}$ is invertible, the result follows.

Ciphertext homomorphism. The function $C(\mathbf{c}, \mathbf{A}, \mathbf{b})$ is defined as follows. For $\mathbf{c} = (g^{\mathbf{a}}, g^u)$, it outputs $(g^{\mathbf{A}^{-T}\mathbf{a}}, g^u \cdot g^{-\mathbf{a}^T\mathbf{A}^{-1}\mathbf{b}})$. We now need to prove that the distribution

$$(\mathrm{pk}, \mathrm{pk}', \mathbf{c}') = ((g^{\mathbf{z}}, g^{-\mathbf{z}^T\mathbf{s}}), (g^{\mathbf{y}}, g^{-\mathbf{y}^T(\mathbf{A}\mathbf{s}+\mathbf{b})}), (g^{r\mathbf{y}}, g^{-r\mathbf{y}^T(\mathbf{A}\mathbf{s}+\mathbf{b})} \cdot w))$$

is computationally indistinguishable from

$$(\mathrm{pk}, \mathrm{pk}', C(\mathbf{c}, \mathbf{A}, \mathbf{b})) = ((g^{\mathbf{z}}, g^{-\mathbf{z}^T\mathbf{s}}), (g^{\mathbf{y}}, g^{-\mathbf{y}^T(\mathbf{A}\mathbf{s}+\mathbf{b})}), (g^{r\mathbf{A}^{-T}\mathbf{z}}, g^{-r\mathbf{z}^T\mathbf{s}} \cdot g^{-r\cdot\mathbf{z}^T\mathbf{A}^{-1}\mathbf{b}} \cdot w)) ,$$

where $\mathbf{z}, \mathbf{y} \xleftarrow{\$} \mathbb{Z}_q^\ell$, $r \leftarrow \mathbb{Z}_q$.

We define a random variable $\mathbf{z}' \leftarrow \mathbb{Z}_q^\ell$. Since $(g^{\mathbf{z}}, g^{r\mathbf{z}})$ is computationally indistinguishable from $(g^{\mathbf{z}}, g^{\mathbf{z}'})$, it follows that $(\mathrm{pk}, \mathrm{pk}', C(\mathbf{c}, \mathbf{A}, \mathbf{b}))$ is computationally indistinguishable from

$$((g^{\mathbf{z}}, g^{-\mathbf{z}^T\mathbf{s}}), (g^{\mathbf{y}}, g^{-\mathbf{y}^T(\mathbf{A}\mathbf{s}+\mathbf{b})}), (g^{\mathbf{A}^{-T}\mathbf{z}'}, g^{-\mathbf{z}'^T\mathbf{s}} \cdot g^{-\mathbf{z}'^T\mathbf{A}^{-1}\mathbf{b}} \cdot w)) .$$

Denoting $\mathbf{y}' = \mathbf{A}^{-T}\mathbf{z}'$, we get

$$((g^{\mathbf{z}}, g^{-\mathbf{z}^T\mathbf{s}}), (g^{\mathbf{y}}, g^{-\mathbf{y}^T(\mathbf{A}\mathbf{s}+\mathbf{b})}), (g^{\mathbf{y}'}, g^{-\mathbf{y}'^T(\mathbf{A}\mathbf{s}+\mathbf{b})} \cdot w)) ,$$

which is computationally indistinguishable from $(\mathrm{pk}, \mathrm{pk}', \mathbf{c}')$ since $(g^{\mathbf{y}}, g^{r\mathbf{y}})$ is computationally indistinguishable from $(g^{\mathbf{y}}, g^{\mathbf{y}'})$. $\square$

B Proof of Lemma 6.3

Let $\mathcal{S}$ be as in the lemma statement, and let $\mathcal{A}$ be an adversary for the $\mathrm{KDM}^{(1)}_{\mathcal{F}_{\mathrm{aff}}}$ security of $\mathcal{E}_{\mathrm{ACPS}}[\mathcal{S}]$. We show that there exists an adversary $\mathcal{B}$ such that

$$\mathrm{LWE}_{q,m,\ell,\chi}[\mathcal{S}]\mathrm{Adv}[\mathcal{B}](\lambda) \geq \mathrm{KDM}^{(1)}_{\mathcal{F}_{\mathrm{aff}}}\mathrm{Adv}[\mathcal{A}, \mathcal{E}_{\mathrm{ACPS}}[\mathcal{S}]](\lambda) - \mathrm{negl}(\lambda) ,$$

where

$$\mathrm{LWE}_{q,m,\ell,\chi}[\mathcal{S}]\mathrm{Adv}[\mathcal{B}](\lambda) = |\Pr[\mathcal{B}(\mathbf{A}, \mathbf{A}\cdot\mathbf{s}+\boldsymbol{\eta}) = 1] - \Pr[\mathcal{B}(\mathbf{A}, \mathbf{u}) = 1]|$$

with $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m\times\ell}$, $\mathbf{s} \xleftarrow{\$} \mathcal{S}$, $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m$ and $\boldsymbol{\eta} \xleftarrow{\$} \chi^m$.

Let $\mathbf{A}$, $\boldsymbol{\eta}$, $\mathbf{s}$ be as above and let $\mathbf{b}^{(0)} = \mathbf{A}\mathbf{s}+\boldsymbol{\eta}$, $\mathbf{b}^{(1)} \xleftarrow{\$} \mathbb{Z}_q^m$. $\mathcal{B}$ gets as input $(\mathbf{A}, \mathbf{b})$ where $\mathbf{b} \in \{\mathbf{b}^{(0)}, \mathbf{b}^{(1)}\}$ and simulates the $\mathrm{KDM}^{(1)}_{\mathcal{F}_{\mathrm{aff}}}$ game for $\mathcal{A}$ as if $(\mathbf{A}, \mathbf{b})$ was a legal public key for $\mathcal{E}_{\mathrm{ACPS}}[\mathcal{S}]$.

Initialize. $\mathcal{B}$ sends $\mathrm{pk} = (\mathbf{A}, \mathbf{b})$ to $\mathcal{A}$, and flips a coin $\xi \xleftarrow{\$} \{0,1\}$.

Queries. Suppose $\mathcal{A}$ makes a query $f_{\mathbf{t},w} \in \mathcal{F}_{\mathrm{aff}}$, $\mathcal{B}$ samples $(\mathbf{u}, v) \xleftarrow{\$} E_{(\mathbf{A},\mathbf{b})}$ and sets $\mathbf{c}_0 = (\mathbf{u}, v) + (-\mathbf{t}\cdot p, w\cdot p)$ and $\mathbf{c}_1 = (\mathbf{u}, v)$. Then, $\mathcal{B}$ returns $\mathbf{c}_\xi$ as an answer to $\mathcal{A}$.

Finish. When $\mathcal{A}$ terminates and returns ξ', $\mathcal{B}$ returns 1 if $\xi' = \xi$ and 0 otherwise.

To analyze $\mathcal{B}$, first consider the case where $\mathbf{b} = \mathbf{b}^{(0)}$, i.e. $(\mathbf{A}, \mathbf{b})$ is a legal public-key for $\mathcal{E}_{\mathrm{ACPS}}[\mathcal{S}]$. In this case, by Lemma 6.1, $\mathcal{B}$ simulates the $\mathrm{KDM}^{(1)}_{\mathcal{F}_{\mathrm{aff}}}$ game up to a negligible statistical distance, and thus

$$\left|\Pr[\mathcal{B}(\mathbf{A}, \mathbf{b}^{(0)}) = 1] - \Pr[\mathcal{A} \text{ wins } \mathrm{KDM}^{(1)}_{\mathcal{F}_{\mathrm{aff}}}]\right| = \mathrm{negl}(\lambda) \ .$$

Next, consider the case where $\mathbf{b} = \mathbf{b}^{(1)}$. In this case, by Lemma 6.2, $\mathbf{c}_0$ and $\mathbf{c}_1$ are within negligible statistical distance, and thus the views of $\mathcal{A}$ where $\xi = 0$ and $\xi = 1$ are within negligible statistical distance. Therefore,

$$\left|\Pr[\mathcal{B}(\mathbf{A}, \mathbf{b}^{(1)}) = 1] - \frac{1}{2}\right| = \left|\Pr[\xi = \xi'] - \frac{1}{2}\right| = \mathrm{negl}(\lambda)$$

and we conclude that

$$\left|\Pr[\mathcal{B}(\mathbf{A}, \mathbf{b}^{(0)}) = 1] - \Pr[\mathcal{B}(\mathbf{A}, \mathbf{b}^{(1)}) = 1]\right| \geq \left|\Pr[\mathcal{A} \text{ wins } \mathrm{KDM}^{(1)}_{\mathcal{F}_{\mathrm{aff}}}] - \frac{1}{2}\right| - \mathrm{negl}(\lambda) \ .$$

Recalling that

$$\mathrm{LWE}_{q,m,\ell,\chi}[\mathcal{S}]\mathrm{Adv}[\mathcal{B}](\lambda) = \left|\Pr[\mathcal{B}(\mathbf{A}, \mathbf{b}^{(0)}) = 1] - \Pr[\mathcal{B}(\mathbf{A}, \mathbf{b}^{(1)}) = 1]\right|$$

and that

$$\mathrm{KDM}^{(1)}_{\mathcal{F}_{\mathrm{aff}}}\mathrm{Adv}[\mathcal{A}, \mathcal{E}_{\mathrm{ACPS}}[\mathcal{S}]](\lambda) = \left|\Pr[\mathcal{A} \text{ wins } \mathrm{KDM}^{(1)}_{\mathcal{F}_{\mathrm{aff}}}] - \frac{1}{2}\right| \ ,$$

the proof is complete. □