On the Circular Security of Bit-Encryption

Ron Rothblum[*]

March 7, 2012

Abstract

Motivated by recent developments in fully homomorphic encryption, we consider the folklore conjecture that every semantically-secure bit-encryption scheme is circular secure, or in other words, that every bit-encryption scheme remains secure even when the adversary is given encryptions of the individual bits of the private-key. We show the following obstacles to proving this conjecture:

1. We construct a public-key bit-encryption scheme that is plausibly semantically secure, but is not circular secure. The circular security attack manages to fully recover the private-key.

   The construction is based on an extension of the Symmetric External Diffie-Hellman assumption (SXDH) from bilinear groups, to ℓ-multilinear groups of order p where $\ell \geq c \cdot \log p$ for some $c > 1$.

   While there do exist ℓ-multilinear groups (unconditionally), for $\ell \geq 3$ there are no known candidates for which the SXDH problem is believed to be hard. Nevertheless, there is also no evidence that such groups do not exist. Our result shows that in order to prove the folklore conjecture, one must rule out the possibility that there exist ℓ-multilinear groups for which SXDH is hard.

2. We show that the folklore conjecture cannot be proved using a black-box reduction. That is, there is no reduction of circular security of a bit-encryption scheme to semantic security of that very same scheme that uses both the encryption scheme and the adversary as black-boxes.

Both of our negative results extend also to the (seemingly) weaker conjecture that every CCA secure bit-encryption scheme is circular secure.

As a final contribution, we show an equivalence between three seemingly distinct notions of circular security for public-key bit-encryption schemes. In particular, we give a general search to decision reduction that shows that an adversary that distinguishes between encryptions of the bits of the private-key and encryptions of zeros can be used to actually recover the private-key.

[*] Department of Computer Science, Weizmann Institute of Science. Email: `ron.rothblum@weizmann.ac.il`. This research was partially supported by the Israel Science Foundation (grant No. 1041/08). Parts of this research were conducted while the author was an intern at Microsoft Research, New England.

1 Introduction

Modern cryptographic applications, both practical and theoretical, have led to the study of increasingly complex types of attacks on encryption schemes. For example, challenge plaintext attacks (CPA) and challenge ciphertext attacks (CCA) extend the classical notion of semantic security [GM84] by allowing an attacker access to encryptions of arbitrary messages of its choice (in the CPA model) and to a decryption oracle (in the CCA model).

A different type of attack that has been recently considered is when the attacker manages to obtain encryptions of messages that are related to the (private) decryption-key. The notion of key dependent message (KDM) security was first considered by Camenisch and Lysyanskaya [CL01] and (independently) by Black *et al.*[BRS02]. Informally, an encryption scheme is KDM secure for a class of functions $\mathcal{F}$ if it is infeasible to distinguish between an oracle that on input $f \in \mathcal{F}$ outputs an encryption of f evaluated on the decryption-key and an oracle that just returns encryptions of zeros.

Perhaps the most basic type of KDM attack is one in which the attacker is just given an encryption of the entire decryption-key. Security with respect to such a KDM attack is also known as *"circular security"* since the key encrypts itself.[1]

While some encryption schemes have been proved to be circular secure under plausible cryptographic assumptions (e.g., [BHHO08, ACPS09]), it is natural to ask whether semantic security actually guarantees circular security. A folklore example shows that this is not the case: given any private-key encryption scheme we can slightly modify the encryption algorithm by checking if the input message is the (symmetric) key itself or not. If not, then the encryption proceeds as usual. But, if the input message equals the key, then the encryption algorithm is modified to output the key in the clear. The resulting scheme is still semantically secure[2] and yet it is not circular secure, since an adversary that gets an encryption of the key trivially breaks security. The counterexample can be easily extended to the public-key setting by having the encryption algorithm check whether a given input message functions as a "good" decryption-key.[3]

The foregoing counterexample shows that, in general, semantic security does not suffice for circular security. Motivated by recent developments in fully homomorphic encryption (see Section 1.2), we restrict our attention to a specific class of encryption schemes - those that encrypt their input bit-by-bit (also called bit-encryption schemes).[4] Thus, we ask whether every *bit-encryption* scheme that is semantically secure is also circular secure. An alternative way to phrase the question is whether every semantically secure (either private-key or public-key) encryption scheme remains secure even if the adversary is given encryptions of the individual bits of the decryption-key (in order, of course).

At this point it is worthwhile to point out two ways in which the counterexample (for full fledged encryption schemes) uses the fact that the encryption algorithm is given the entire decryption-key as its message:

1. It is easy to identify when the decryption-key is given as the input message to the encryp-

[1] Circular security may also refer to larger key cycles were there are t keys arranged in a directed cycle and the adversary sees encryptions under every key of its next neighbor's key. We only consider the case $t = 1$.

[2] Semantic security follows from the fact that the probability that the message (which is selected before the keys) equals the key is negligible.

[3] The (public-key) encryption algorithm can do so by encrypting sufficiently many random messages and checking whether the given input message (used as a decryption-key) correctly decrypts these ciphertexts.

[4] We assume that the encryption algorithm does not maintain a state between executions. Note that the folklore counterexample for full fledged encryption can be adapted to *stateful* bit-encryption schemes by having the encryption algorithm record its last n (single-bit) messages in a buffer (where n is the length of the decryption-key), and outputting the decryption-key in the clear whenever the buffer equals the decryption-key.

tion algorithm (trivially in the private-key setting and almost as easily in the public-key setting); and

2. In the semantic security setting, the event that the message equals the decryption-key is sufficiently rare that we can modify the encryption algorithm to handle this event in a special way without jeopardizing security.

In the case of bit-encryption schemes both properties no longer hold and constructing a counterexample seems to be more difficult.[5] In fact, the above has led to a folklore conjecture, which we call the bit-encryption conjecture, that *every* secure bit-encryption scheme (either private-key or public-key) is in fact circular secure. Let us state this as:

Conjecture 1 (Bit-Encryption Conjecture)**.** *Every semantically secure public-key bit-encryption scheme is circular secure.*

The focus of this work is to show obstacles to proving the validity of this conjecture. Focusing on the public-key case only strengthens our negative results since every public-key scheme is also a private-key scheme. (In Section 1.4 we also discuss the (seemingly) weaker conjecture that every CCA secure bit-encryption scheme is circular secure.)

1.1 Our Results

We address the question of circular security for bit-encryption schemes and show the following results:

A circular insecure bit-encryption scheme based on ℓ-multilinear maps. We construct a (plausibly) semantically secure public-key encryption scheme for which, given encryptions of the bits of the decryption-key, it is possible to fully recover the decryption-key (i.e., the strongest type of attack). The security of our construction is based on an extension of the Symmetric External Diffie-Hellman (SXDH) assumption (see [BGdMM05, FGHP09]) to multilinear groups, which we describe next.

An ℓ-multilinear map is a (non-degenerate) mapping $e : G_1 \times \cdots \times G_\ell \to G_T$ where $G_1, \ldots, G_\ell$ and G_T are cyclic groups of prime order p, such that for every $g_1 \in G_1, \ldots, g_\ell \in G_\ell$, every $i \in [\ell]$ and $a \in \mathbb{Z}_p$ it holds that

$$e(g_1, \ldots, g_i^a, \ldots, g_\ell) = e(g_1, \ldots, g_\ell)^a.$$

Recall that, informally, the Decisional Diffie Hellman (DDH) assumption is said to hold in the cyclic group G if it is infeasible to distinguish between g, g^a, g^b, g^{ab} and g, g^a, g^b, g^c where g is a generator of G and a, b and c are random exponents. The standard SXDH assumption extends the DDH assumption to 2-multilinear (a.k.a bilinear) groups by stating that there exist groups (G_1, G_2) equipped with a bilinear map for which the DDH assumption holds (separately) for each one of the groups G_1 and G_2. We further extend the SXDH assumption by assuming that there exist ℓ-multilinear groups for which DDH is hard in each one of the ℓ groups. For our result to hold we need $\ell \geq c \cdot \log p$ for some $c > 1$.

Since, for $\ell > 2$, we do not have candidate ℓ-multilinear groups for which we conjecture SXDH to be hard, we do not interpret our construction as a counterexample, but rather as an obstacle to proving the bit-encryption conjecture (Conjecture 1). Our construction shows that

[5] In fact, for the very same reasons, even constructing an encryption scheme for logarithmically long messages that is semantically secure but circular *insecure* seems to be difficult. We note that our negative results extend also to this case but in this work we only discuss the single bit case.

in order to prove that every semantically secure bit-encryption scheme is circular secure one would have to rule out the existence of ℓ-multilinear groups for which SXDH is hard.

The possibility of constructing ℓ-multilinear group schemes for which discrete log is hard was previously considered by Boneh and Silverberg [BS03], who showed cryptographic applications of multilinear maps as well as difficulties in constructing such group schemes based on known techniques in algebraic geometry. We note that [BS03] only considered the special case of $G_1 \equiv \ldots \equiv G_\ell$ and the hardness of discrete log for G_1 (in fact, if $G_1 \equiv \ldots \equiv G_\ell$ then SXDH becomes trivially easy[6]).

Impossibility of black-box reductions. We show that a black-box reduction cannot be used to prove the bit-encryption conjecture. Our black-box impossibility result differs from standard black-box impossibility results in that we do not consider the possibility of *constructing* a circular secure bit-encryption from any semantically-secure bit-encryption but rather the question of whether every semantically secure bit-encryption is *by itself* already circular secure.

In other words, we prove that there cannot exist a general black-box reduction that transforms any circular security attack into a semantic security attack. By black-box we mean that the reduction uses both the attack and the primitive (in our case the encryption scheme) in a black-box manner (for a discussion of different types of black-box separations, see [RTV04]).

From indistinguishability to key-recovery. We show an equivalence between three natural notions of circular security for public-key bit-encryption schemes. In all three scenarios we give the adversary access to an oracle that on input i returns an encryption of the i-th bit of the decryption-key. We refer to this oracle as the KDM oracle. The three security notions differ in the task that a hypothetical adversary, which has access to the KDM oracle, has to accomplish in order to be deemed successful (i.e., break security). We consider the following possible tasks:

1. The adversary needs to fully recover the decryption-key.

2. The adversary gets as input an encryption of a random bit and needs to guess the value of this bit.

3. The adversary is given access to either the KDM oracle or an oracle that always returns encryptions of 0 and needs to distinguish in which of the two cases it is. This is the standard notion of circular security as defined in [CL01, BRS02].

We show that the three foregoing notions are actually equivalent. In particular, this result implies a general search to decision reduction that transforms any circular security distinguisher into an adversary that, given access to the KDM oracle, can fully recover the decryption-key d. (In contrast, in the setting of semantic security, finding the key can be a much harder task than recovering the message from the ciphertext.)

1.2 Connection to Fully Homomorphic Encryption and Full KDM Security

Other than being an interesting and natural question on its own, the question of circular security for bit-encryption schemes is further motivated by recent breakthroughs in the construction of fully homomorphic encryption schemes (FHE) and fully KDM secure encryption schemes.

[6] Using the fact that the groups are equals, we can solve DDH in G_1. Specifically, given $g, g^a, g^b, g^c \in G_1$ just compare $e(g^a, g^b, g, \ldots, g)$ and $e(g^c, g, \ldots, g)$, where we use the fact that $g^b \in G_2 = G_1$. If $c = ab$ then equality holds but if c is random then the two values are different with overwhelming probability.

Fully Homomorphic Encryption. Informally, an FHE is an encryption scheme for which given an encryption of a message m and *any* circuit C, one can compute an encryption of $C(m)$ without knowing the decryption-key.

Gentry [Gen09] constructed the first FHE and gave a general technique called bootstrapping for the construction of FHE schemes. Gentry's idea is to first construct an encryption scheme that is somewhat homomorphic (that is, homomorphic with respect to some limited class of circuits), and then, using the bootstrapping technique, to transform it into an FHE. The bootstrapping technique inherently uses the assumption that the underlying somewhat homomorphic encryption is circular secure.[7] Since most of these schemes are bit encryption schemes and their circular security is only conjectured and not proved (based on their semantic security), the question of circular security for bit-encryption is especially important for the construction of secure FHE. In particular, proving the bit-encryption conjecture would establish the existence of an FHE based solely on (say) the hardness of the learning with errors (LWE) problem (see [BV11]).

Our KDM equivalence theorem for bit-encryption (see end of Section 1.1) is also of particular interest to the current candidate FHE schemes. As alluded to above, the theorem implies that a KDM distinguisher can be used to construct an attacker that *given access to the KDM oracle* actually finds the decryption-key. However, for the current candidate fully homomorphic bit-encryption schemes, the KDM oracle can actually be simulated using only the public-key.[8] Thus, the equivalence theorem gives a generic (and simple) search to decision reduction for these schemes that transforms any attack that breaks semantic security into an attack that finds the decryption-key (without using an external KDM oracle).

Full KDM Security from Semantic Security. An additional motivation for the study of the circular security of bit-encryption schemes arises from the recent work of Applebaum [App11] (following [BHHI10, BGK11]) who showed an amplification theorem for KDM security. Specifically, [App11] showed that an encryption scheme that is KDM secure for any fixed class of polynomial-size circuits, can be constructed from an encryption scheme that is KDM secure only with respect to the class of projections and negation of projections (i.e., any function f of the form $f(d) = d_i$ or $f(d) = 1 - d_i$). Thus, proving a slightly stronger variant of the bit-encryption conjecture would imply that semantic security is a sufficient assumption for the construction of a very strong form of KDM security.

1.3 Comparison of our Black-Box Impossibility Result with [HH09]

Haitner and Holenstein [HH09] gave the following two black-box impossibility results regarding the construction of KDM secure encryption schemes:

1. There exists no fully black-box reduction from an encryption scheme that is secure for a class of KDM functions to a collection of trapdoor permutations, if the class of KDM functions contains a collection of poly(n)-wise independent hash functions.

[7] Actually, there are two variants of the bootstrapping technique. The one that we refer to assumes circular security and constructs an FHE. The other variant does not assume circular security but only achieves *leveled* FHE (i.e., an encryption scheme that is homomorphic with respect to any circuit of some apriori fixed depth) and also expands the public-key by a multiplicative factor that is linear in the depth of supported circuits.

[8] This follows from the facts that (1) the public-key of these schemes actually contains encryptions of the bits of the decryption-key (for bootstrapping), and (2) ciphertexts can be re-randomized. An oracle query for the i-th bit of the decryption-key can be simulated by re-randomizing the ciphertext in the public-key that is an encryption of the i-th bit of the decryption-key.

2. There exists no fully black-box construction of an encryption scheme that is secure for a class of KDM functions to any cryptographic game, if the reduction treats the KDM functions as a black-box.

Circular security of bit-encryption corresponds exactly to KDM security with respect to the class of projection functions. The results of [HH09] do not apply to this class of KDM functions because (1) the class of projection functions does not include a collection of poly(n)-wise independent hash functions, and (2) we consider a concrete class of KDM functions and the reduction may make arbitrary (non-black-box) use of these functions. Therefore our black-box impossibility is not covered by the results of [HH09].

We also wish to point out that [HH09] ruled out a wide range of *constructions* of KDM secure encryption schemes. Our black-box impossibility result, on the other hand, only rules out the possibility of proving (via a black-box reduction) that every semantically secure bit-encryption scheme is also circular secure *without any modification to the encryption scheme.* In other words, we do not rule out the possibility that there exists a black-box *construction* of a circular secure bit-encryption scheme from any semantically secure bit-encryption scheme.

1.4 Chosen Ciphertext Security vs. Circular Security

Recall that an encryption scheme is CCA-2 secure if it is semantically secure even when the attacker has access to a decryption oracle that decrypts any ciphertext other than the challenge ciphertext.

Since we show difficulties to proving that every semantically secure bit-encryption is circular secure, it is natural to ask whether a stronger notion of security, such as CCA security, might instead suffice. We first note that our black-box impossibility result extends also to this case. That is, we show that there is no blackbox reduction of circular-security even to CCA-2 security.

Actually, assuming the existence of doubly-enhanced trapdoor permutations, the conjecture that every CCA bit-encryption scheme is circular-secure is equivalent to the bit encryption conjecture. This equivalence follows from the fact that the Naor-Yung paradigm [NY90] transforms a semantically secure but circular insecure scheme into a CCA secure but circular *insecure* one.[9] Using this observation we can extend our construction of a circular-insecure bit-encryption scheme (based on multilinear SXDH, see Section 3) to a CCA-2 secure but circular-insecure bit-encryption scheme (assuming, in addition to multilinear SXDH, the existence of doubly-enhanced trapdoor permutations).[10]

Remark. We also mention that the converse direction that asks whether every circular secure bit-encryption scheme is also CCA secure is in fact false (assuming that there exist circular secure bit-encryption schemes at all). For example, taking any circular secure scheme and modifying it by adding to the public-key an encryption of the decryption-key, yields a scheme that is circular secure but is not even CCA-1 secure.

[9] Recall that the Naor-Yung paradigm consists of a double encryption of the plaintext using independent keys and a non-interactive zero-knowledge (NIZK) proof of consistency. A circular security attack on the underlying scheme immediately translates into a circular security attack on the constructed CCA secure scheme. Note that the Naor-Yung transformation can be made to achieve not only CCA-1 security but even CCA-2 security (see [Sah99] or [Lin06]).

[10] We note that this equivalence does not directly imply the extension of our black-box result to the CCA case because the Naor-Yung transformation makes non black-box use of the encryption scheme. Instead we prove the extension of the black-box result directly (without even assuming the existence of doubly-enhanced trapdoor permutations).

Organization

In Section 2 we define KDM security and the cryptographic assumptions that we will use. In Section 3 we present our "multilinear map" based circular insecure bit-encryption scheme. In Section 4 we prove the equivalence of three notions of KDM security. Finally, in Section 5, we present our black-box impossibility result.

2 Preliminaries

We denote by $x \in_R S$ a random variable x that is uniformly distributed in the set S.

2.1 Public-Key Encryption

A public-key encryption scheme consists of three probabilistic polynomial-time algorithms $KeyGen$, Enc and Dec. The key generation algorithm, $KeyGen$, when given as input a security parameter 1^n, outputs a pair (e, d) of poly(n)-bit long encryption and decryption keys. The encryption algorithm, Enc, on input an encryption-key e and a message $m \in \{0, 1\}^*$, outputs a ciphertext c, whereas the decryption algorithm, Dec, when given the ciphertext c and the decryption-key d, outputs m. We say that the encryption scheme is correct if for every message m and every valid key-pair (e, d) it holds that $Dec_d(Enc_e(m)) = m$.

In this work we restrict our attention to bit-encryption scheme (i.e., $m \in \{0, 1\}$). We first define the classical notion of semantic security restricted to single-bit encryption:

Definition 2. *A public-key bit-encryption scheme is* semantically-secure *if for every probabilistic polynomial-time adversary A it holds that*

$$\Pr_{\substack{(e,d)\leftarrow KeyGen(1^n)\\ b\in_R\{0,1\}}}[A(e, Enc_e(b)) = b] < \frac{1}{2} + \text{neg}(n).$$

Next, we define chosen ciphertext attack (CCA) security for bit-encryption. In this work we only consider the stronger notion of CCA-2:

Definition 3. *A public-key bit-encryption scheme is* CCA-2 secure *if for every probabilistic polynomial-time adversary A it holds that*

$$\Pr_{\substack{(e,d)\leftarrow KeyGen(1^n)\\ b\in_R\{0,1\}}}\left[A^{Dec'_d}(e, Enc_e(b)) = b\right] < \frac{1}{2} + \text{neg}(n)$$

where Dec'_d is an oracle that on input c returns $\perp$ if $c = Enc_e(b)$ is the challenge ciphertext and otherwise returns $Dec_d(c)$ (i.e., decrypts the ciphertext).

2.2 KDM and Circular Security for Bit-Encryption

To model KDM security we need to specify what information is given to the adversary and what it means for the adversary to break security. The former is the simpler of the two - we simply give the adversary access to an oracle (henceforth called the KDM oracle) that on input i returns an encryption of the i-th bit of the decryption-key. Formally, for a pair (e, d) of encryption and decryption keys, we define an oracle $O_{e,d}(i)$ which on input $i \in [|d|]$ returns $Enc_e(d_i)$.

Turning to the second part of the definition, we consider three possible ways in which an adversary can break security. The strongest type of attack (which corresponds to the weakest definition of security) that we consider is full key recovery. Security against this type of attack means that no efficient adversary, which gets encryptions of the individual bits of the decryption-key, can find the entire decryption-key. Using the definition of the oracle $O_{e,d}$ we can define circular security of bit-encryption with respect to key-recovery:

Definition 4. *A public-key bit-encryption scheme* $(KeyGen, Enc, Dec)$ *is* circular secure with respect to key recovery *if for every probabilistic polynomial-time oracle machine A it holds that*

$$\Pr_{(e,d)\leftarrow KeyGen(1^n)}\left[A^{O_{e,d}}(e) = d\right] < \mathrm{neg}(n).$$

It is worth noting that, in contrast to the semantic security setting, in the KDM setting the decryption-key is information theoretically determined and therefore there is at least some hope to recover the actual decryption-key used by the scheme.[11]

Next, we consider an adversary that is given an encryption of a random bit, as well as access to the KDM oracle, and needs to guess the value of the bit:

Definition 5. *A public-key bit-encryption scheme is* circular secure with respect to message recovery *if for every probabilistic polynomial-time oracle machine A it holds that*

$$\Pr_{\substack{(e,d)\leftarrow KeyGen(1^n),\\ b\in_R\{0,1\}}}\left[A^{O_{e,d}}(e, Enc_e(b)) = b\right] < \frac{1}{2} + \mathrm{neg}(n).$$

Lastly, we consider the standard definition of circular security as put forth by [CL01, BRS02]. Their definition requires that if be infeasible for an adversary to distinguish between the KDM oracle and an “all zeros” oracle that always returns encryptions of 0. Formally, for an encryption-key e, we define J_e to be an oracle that on input i just returns $Enc_e(0)$ (i.e., an encryption under e of the bit 0). In contrast to the two prior definitions, indistinguishability of oracles does not inherently imply semantic security and therefore we explicitly add this requirement.

Definition 6. *A semantically-secure public-key bit-encryption scheme is* circular secure with respect to indistinguishability of oracles *if for every probabilistic polynomial-time oracle machine A it holds that*

$$\left|\Pr_{(e,d)\leftarrow KeyGen(1^n)}\left[A^{O_{e,d}}(e) = 1\right] - \Pr_{(e,d)\leftarrow KeyGen(1^n)}\left[A^{J_e}(e) = 1\right]\right| < \mathrm{neg}(n).$$

In Section 4 we show that the three notions of circular security presented above are actually equivalent.

2.3 Hardness assumptions in bilinear and ℓ-multilinear groups

We first define bilinear and ℓ-multilinear maps and then define the computational assumptions that we use.

An ℓ-multilinear map is a non-degenerate[12] function $e : G_1 \times \cdots \times G_\ell \to G_T$, where $G_1, \ldots, G_\ell, G_T$ are cyclic groups of prime order p such that for every $g_1 \in G_1, \ldots, g_\ell \in G_\ell$, every $i \in [\ell]$ and $a \in \mathbb{Z}_p$, it holds that:

$$e(g_1, \ldots, g_i^a, \ldots, g_\ell) = e(g_1, \ldots, g_\ell)^a.$$

[11] In the semantic security model, there may be many decryption keys corresponding to the same encryption-key and a semantic security adversary (which only has access to functions of the encryption-key) cannot hope to always find the particular decryption-key being used.

[12] Where by degenerate we mean a function that maps all inputs to the identity element of G_T.

An ℓ-multilinear group scheme is an algorithm that for every security parameter n produces a description of $\ell+1$ groups of order p (where p is an n-bit prime) together with an efficiently computable ℓ-multilinear map that maps the first ℓ groups to the $(\ell+1)$-th group:

Definition 7. *Let $\ell = \ell(n)$ be a polynomially bounded function. An* ℓ-multilinear group scheme *is a probabilistic polynomial-time algorithm GS that on input 1^n outputs the parameters $params = (p, (G_1, \ldots, G_\ell, G_T), (g_1, \ldots, g_\ell, g_T), e)$ where $2^{n-1} < p < 2^n$ is an n-bit prime, $G_1, \ldots, G_\ell$ and G_T are concise descriptions of $\ell+1$ groups of order p (that allow efficient evaluation of the group operation), with the respective generators $g_1, \ldots, g_\ell, g_T$ and $e : G_1 \times \cdots \times G_\ell \to G_T$ is a concise description of an efficiently computable ℓ-multilinear map.*

For every ℓ there exist trivial examples of ℓ-multilinear group schemes. However, our computational hardness assumptions do not hold for these trivial examples.[13] In fact, for $\ell \geq 3$ we do not know of a candidate ℓ-multilinear group scheme for which the discrete log problem is believed to be hard (in any of the groups). Nevertheless, there is also no negative evidence that such group schemes do not exist. For $\ell \leq 2$ there do exist candidate group schemes for which discrete log is conjectured to be hard (discussed next).

Computational Assumptions. Loosely speaking, the DDH assumption for a cyclic group G states that the distributions (g, g^a, g^b, g^{ab}) and (g, g^a, g^b, g^c) are computationally indistinguishable, where g is a generator of G and a, b and c are random exponents. The SXDH assumption (introduced by [BGdMM05, FGHP09]) extends DDH to 2-multilinear (a.k.a bilinear) groups by assuming that there exist groups G_1, G_2 equipped with a bilinear map such that the DDH assumption holds for both G_1 and G_2 (separately). We further extend SXDH to the ℓ-multilinear SXDH assumption which states that there exists an ℓ-multilinear group scheme for which DDH is hard for all ℓ groups $G_1, \ldots, G_\ell$. Note that 1-multilinear SXDH corresponds exactly to DDH and that 2-multilinear SXDH corresponds to the standard SXDH assumption. We emphasize that we only have candidate group schemes for which the ℓ-multilinear SXDH assumption is conjectured to hold for $\ell \leq 2$ (see [BGdMM05, FGHP09]).

Definition 8. *The* ℓ-multilinear SXDH assumption *states that there exists an ℓ-multilinear group scheme GS such that for every function $i \colon \mathbb{N} \to \mathbb{N}$ for which $i(n) \in [\ell(n)]$, the following ensembles are computationally indistinguishable:*

1. $\{params, i(n), g_{i(n)}^a, g_{i(n)}^b, g_{i(n)}^{ab}\}_{n \in \mathbb{N}}$*; and*
2. $\{params, i(n), g_{i(n)}^a, g_{i(n)}^b, g_{i(n)}^c\}_{n \in \mathbb{N}}$

where $a, b, c \in_R \mathbb{Z}_p$ and $params \stackrel{\text{def}}{=} (p, (G_1, \ldots, G_\ell, G_T), (g_1, \ldots, g_\ell, g_T), e)$ is distributed as $GS(1^n)$.

3 A Circular Insecure Bit-Encryption Scheme

In this section we show a construction of a bit-encryption scheme $(KeyGen, Enc, Dec)$ that is (plausibly) semantically secure but is not circular secure. In Section 3.1 we present the construction. In Section 3.2 we prove that the construction is correct and semantically secure

[13] A trivial example of an ℓ-multilinear group scheme is when $G_1, \ldots, G_\ell$ are all the *additive* group mod p. Since exponentiation in the additive group corresponds to modular multiplication, being multilinear means that for every $a, z_1, \ldots, z_\ell \in \mathbb{Z}_p$ it holds that $e(z_1, \ldots, a \cdot z_i, \ldots, z_\ell) = a \cdot e(z_1, \ldots, z_\ell)$. Hence, the mapping $e(z_1, \ldots, z_\ell) = \prod_{i=1}^{\ell} z_i \bmod p$ is a multilinear map for these groups. Note however that discrete log in the additive group is equivalent to modular division and can therefore be efficiently computed.

(based on the hardness of ℓ-multilinear SXDH, for $\ell \geq c \cdot \log p$ for some constant $c > 1$). In Section 3.3 we use the multilinear map to show a circular security attack on the scheme.

Notation. For a matrix X, we let $X[i,j]$ denote the (i,j)-th entry of X.

3.1 The Encryption Scheme

Let GS be any ℓ-multilinear group scheme (as in Definition 7).

Construction 9. *Consider the following public-key bit-encryption scheme* $(KeyGen, Enc, Dec)$*:*

$\underline{KeyGen(1^n)}$

1. *Invoke the group scheme algorithm to obtain params* $\leftarrow GS(1^n)$ *(where params* $= (p, (G_1, \ldots, G_\ell, G_T), (g_1, \ldots, g_\ell, g_T), e)$*).*
2. *Select* $X \in_R \mathbb{Z}_p^{2\times\ell}$ *(i.e., a* $2 \times \ell$ *matrix with random entries in* $\mathbb{Z}_p$*).*
3. *Set* $U = \begin{bmatrix} g_1^{X[0,1]} & g_2^{X[0,2]} & \cdots & g_\ell^{X[0,\ell]} \\ g_1^{X[1,1]} & g_2^{X[1,2]} & \cdots & g_\ell^{X[1,\ell]} \end{bmatrix}$.
4. *Select* $s \in_R \{0,1\}^\ell$ *and set* $\alpha = \sum_{i=1}^{\ell} X[s_i, i] \bmod p$.
5. *The (public) encryption-key is* $(params, U, \alpha)$ *and the (private) decryption-key is* (X, s)*.*

$\underline{Enc_{(params,U,\alpha)}(\sigma)}$ *(where* $\sigma \in \{0,1\}$*)*

1. *Select at random* $r_1, \ldots, r_\ell \in_R \mathbb{Z}_p$*.*
2. *Output* $(g_1^{r_1}, (U[\sigma,1])^{r_1}), \ldots, (g_\ell^{r_\ell}, (U[\sigma,\ell])^{r_\ell})$*.*

$\underline{Dec_{(X,s)}((c_1, d_1), \ldots, (c_\ell, d_\ell))}$

1. *If* $c_1^{X[0,1]} = d_1$ *output 0 and otherwise output 1.*

Before proceeding to the proof of correctness and security, we wish to highlight a few points. First, we note that both α and s are not used by the encryption or decryption algorithms and seem unneeded. Second, we note that (ignoring the presence of α in the public-key) even by setting $\ell = 1$ we obtain a secure encryption scheme (under DDH) and it is not clear why we need a larger ℓ (recall that we need $\ell >> \log p$).

The reason for the existence of α and s is (solely) to help the KDM attacker whereas the large value of ℓ helps maintain semantic security despite the fact that α is revealed in the public-key.[14] The key idea is that in the semantic security setting, an attacker has essentially no information about s (because ℓ is sufficiently large that α looks random) whereas, in the KDM setting, the attacker can obtain additional information about s (specifically encryptions of the bits of s) and can use this additional information to verify that α is consistent with s.

The above gives us a way to distinguish between the KDM oracle and the all zeros oracle thereby breaking circular security with respect to indistinguishability of oracles. (Using the results of Section 4 this attack can be transformed into a full key-recovery attack.)

[14] Note that when using small values of ℓ (in particular using $\ell = 1$), the fact that α is revealed in the public-key makes the scheme totally insecure.

3.2 Correctness and Semantic Security of Construction 9

In this section we show that Construction 9 is both correct (i.e., the decryption of an encryption recovers the original message) and semantically secure.

Correctness. Consider a pair of encryption and decryption keys $((params, U, \alpha), (X, s))$ and let $((c_1, d_1), \ldots, (c_\ell, d_\ell))$ be an encryption of a bit $\sigma \in \{0,1\}$. If $\sigma = 0$ then $d_1 = c_1^{X[0,1]}$ and the ciphertext decrypts correctly to 0. If $\sigma = 1$ then $d_1 = c_1^{X[1,1]}$ and therefore, except with negligible probability, $d_1 \neq c_1^{X[0,1]}$. Hence, the ciphertext decrypts correctly to 1 (except with negligible probability).

Note that we can easily eliminate the negligible decryption error by sampling X from a statistically close distribution in which $X[0,1] \neq X[1,1]$.

Semantic Security. An adversary attempting to break the semantic security of Construction 9 sees the public-key $(params, U, \alpha)$ and an encryption of a random bit σ and needs to find σ (with non-negligible advantage). As a first step to proving semantic security, consider replacing the selection of α in the key generation process by a uniform random $\beta \in_R \mathbb{Z}_p$ (instead of the sum of a random subset). We show that view of the adversary in the resulting scheme is statistically close to its view in the original scheme.

Claim 10. *For every $n \in \mathbb{N}$, the distribution (X, α) and (X, β) are $\frac{1}{2}\sqrt{\frac{p}{2^\ell}}$-close in statistical distance where p, X and α are distributed as in the key generation process and β is an independent random number in $\mathbb{Z}_p$.*

Proof. Follows directly from the leftover hash lemma[15] [HILL99] by observing that the family of functions $\{H_X : \{0,1\}^\ell \to \mathbb{Z}_p\}_X$ defined as $H_X(s) = \sum_{i=1}^{\ell} X[s_i, i] \bmod p$ is a universal hash function family. □

Since an adversary attempting to break the security of the scheme does not see *any* information on s, it is enough to prove that the scheme is secure if α were replaced by $\beta \in_R \mathbb{Z}_p$. Note that we crucially use the fact that *information theoretically* s is completely undetermined by the view of the adversary. In contrast, in the KDM setting, where the adversary can see encryptions of the bits of s this fact no longer holds and our attack exploits this to break the KDM security of the scheme (see Section 3.3).

Lemma 11. *Assuming the ℓ-multilinear SXDH assumption, for $\ell \geq c \cdot \log p$ for some $c > 1$, it holds that for every $\sigma \in \{0,1\}$, the following two distributions are computationally indistinguishable:*

- *$params, U, \beta, (g_1^{r_1}, (U[\sigma,1])^{r_1}), \ldots, (g_\ell^{r_\ell}, (U[\sigma,\ell])^{r_\ell})$; and*
- *$params, U, \beta, (g_1^{r_1}, g_1^{s_1}), \ldots, (g_\ell^{r_\ell}, g_\ell^{s_\ell})$*

where $params$ and U are selected as in the key-generation process and $\beta, r_1, \ldots, r_\ell, s_1, \ldots, s_\ell \in_R \mathbb{Z}_p$.

Note that the by Claim 10 (and since $\ell > c \cdot \log p$ for some constant $c > 1$), the first distribution is statistically close to a public-key of the scheme together with an encryption of a bit σ and the second distribution contains no information about σ. Hence, the semantic security of the scheme follows directly from Lemma 11.

[15] A simplified version of the leftover hash lemma states that if h is selected at random from a universal hash function family from $\mathcal{X}$ to $\mathcal{Y}$ then the distribution $(h, h(x))$ and the distribution (h, y) are $\frac{1}{2}\sqrt{\frac{|\mathcal{Y}|}{|\mathcal{X}|}}$-close, where x and y are uniformly distributed in $\mathcal{X}$ and $\mathcal{Y}$ (see, e.g., [Gol08, Appendix D]).

Proof. The straightforward proof is by a hybrid argument and an application of the ℓ-multilinear SXDH assumption. Details follow.

For a bit $\sigma \in \{0,1\}$, let $H_n^{(i)}$ denote the hybrid distribution:

$$H_n^{(i)} \overset{\text{def}}{=} params, U, \beta, (g_1^{r_1}, (U[\sigma,1])^{r_1}), \ldots, (g_i^{r_i}, (U[\sigma,i])^{r_i}), (g_{i+1}^{r_{i+1}}, g_{i+1}^{s_{i+1}}), \ldots, (g_\ell^{r_\ell}, g_\ell^{s_\ell})$$

Note that the two extreme hybrids $H_n^{(0)}$ and $H_n^{(\ell)}$ correspond exactly to the distributions of Lemma 11. Suppose toward a contradiction that there exists a probabilistic polynomial-time distinguisher D that distinguishes between the two distributions. In other words:

$$\left|\Pr[D(H_n^{(0)}) = 1] - \Pr[D(H_n^{(\ell)}) = 1]\right| > \frac{1}{\text{poly}(n)} \tag{1}$$

Therefore, there must exist $j \in [\ell]$ such that D distinguishes between the two neighboring hybrid distributions $H_n^{(j-1)}$ and $H_n^{(j)}$:

$$\left|\Pr[D(H_n^{(j-1)}) = 1] - \Pr[D(H_n^{(j)}) = 1]\right| > \frac{1}{\text{poly}(n)} \tag{2}$$

We use D to break the ℓ-multilinear SXDH assumption. To do so we need to show a polynomial-time distinguisher D' and a function $i : \mathbb{N} \to \mathbb{N}$ (with $i(n) \in [\ell(n)]$) for which

$$\left| \Pr\left[D'(params, i(n), g_{i(n)}^a, g_{i(n)}^b, g_{i(n)}^{ab}) = 1 \,\middle|\, params \leftarrow GS(1^n),\ a, b \in_R \mathbb{Z}_p\right] \right.$$
$$\left. - \Pr\left[D'(params, i(n), g_{i(n)}^a, g_{i(n)}^b, g_{i(n)}^{c}) = 1 \,\middle|\, params \leftarrow GS(1^n),\ a, b, c \in_R \mathbb{Z}_p\right] \right| < \text{neg}(n)$$

(That is, D' gets as input $(params, i(n), g_{i(n)}, g_{i(n)}^a, g_{i(n)}^b, g_{i(n)}^c)$ and needs to decide whether $c = ab$ or is an independent random number in $\mathbb{Z}_p$.)

Consider the function i that for every n in the infinite set of n's for which D distinguishes between the $(j-1)$-th hybrid and the j-th hybrid just equals j (and for other n equals some arbitrary value). For sake of simplicity we write $i \overset{\text{def}}{=} i(n)$ in the following.

In order to distinguish, D' first generates a matrix U as in the key generation of the scheme with respect to *params* (which are part of its input) by taking random exponents of $g_1, \ldots, g_\ell$. From U, the distinguisher generates a matrix U' which is the same as U' except that the (σ, i)-th entry of U is replaced with g_i^a (note that U' is still distributed identically to a matrix U of the encryption-scheme). The distinguisher D' then selects $\beta, r_1, \ldots, r_\ell, s_{i+1}, \ldots, s_\ell \in_R \mathbb{Z}_p$ and outputs $D(params, U', \beta, \mu)$ where

$$\mu = \left(g_1^{r_1}, (U'[\sigma,1])^{r_1}\right), \ldots, \left(g_{i-1}^{r_{i-1}}, (U'[\sigma,i-1])^{r_{i-1}}\right), \left(g_i^b, g_i^c\right), \left(g_{i+1}^{r_{i+1}}, g_{i+1}^{s_{i+1}}\right), \ldots, \left(g_\ell^{r_\ell}, g_\ell^{s_\ell}\right).$$

Observe that in that case $c = ab$ the input to D is identically distributed to $H_n^{(i)}$ and in the case that c is random in $\mathbb{Z}_p$ the input is identically distributed to $H_n^{(i-1)}$. Thus, for the infinite set for which Eq. (2) holds, D' distinguishes between the two distributions. □

Remark. We can extend Construction 9 also to the case $\ell = c \cdot \log p$ for $0 < c < 1$, assuming that the subset sum assumption holds. Specifically, using the fact that subset sum is a pseudorandom generator (if it is one-way, see [IN89]), we can replace the use of the leftover hash lemma in the proof of semantic security with the subset sum assumption over $\mathbb{Z}_p$ (assuming that subset sum is one-way for the specific setting of parameters).

3.3 The KDM Attack

We show a distinguisher that breaks the *circular security with respect to indistinguishability of oracles* (Definition 6) of Construction 9. Using Theorem 12, we can obtain a KDM attack that breaks *circular security with respect to key recovery* (Definition 4).

Our distinguisher gets as input a public-key and has access to either the KDM oracle that on input i returns an encryption of the i-th bit of the decryption-key or to the all-zeros oracle that always returns an encryption of 0. The goal of the distinguisher is to distinguish between the two cases.

Consider the following distinguisher which has access to an alleged KDM oracle and gets as input an encryption-key $(params, U, \alpha)$:

1. For $i = 1, \dots, \ell$:
   (a) Query the oracle for an encryption $((c_1, d_1), \dots, (c_\ell, d_\ell))$ of s_i (the i-th bit of s).
   (b) Set $y_i = c_i$ and $z_i = d_i$.
2. If $e(y_1, \dots, y_\ell)^\alpha \equiv_p \prod_{i=1}^{\ell} e(y_1, \dots, y_{i-1}, z_i, y_{i+1}, \dots, y_\ell)$ then output 1 and otherwise output 0 (where $\equiv_p$ denotes congruence mod p).

We first show that when using the KDM oracle, the distinguisher always outputs 1. Indeed, in this case $y_i = g_i^{r_i}$ and $z_i = g_i^{r_i \cdot X[s_i,i]}$. Therefore,

$$\begin{aligned}
\prod_{i=1}^{\ell} e(y_1, \dots, y_{i-1}, z_i, y_{i+1}, \dots, y_\ell) &\equiv_p \prod_{i=1}^{\ell} e(g_1^{r_1}, \dots, g_{i-1}^{r_{i-1}}, g_i^{r_i X[s_i,i]}, g_{i+1}^{r_{i+1}}, \dots, g_\ell^{r_\ell}) \\
&\equiv_p \prod_{i=1}^{\ell} e(g_1^{r_1}, \dots, g_\ell^{r_\ell})^{X[s_i,i]} \\
&\equiv_p e(g_1^{r_1}, \dots, g_\ell^{r_\ell})^{\sum_{i=1}^{\ell} X[s_i,i] \bmod p} \\
&\equiv_p e(y_1, \dots, y_\ell)^\alpha
\end{aligned}$$

and so the distinguisher outputs 1 in this case.

Next, consider the case that the distinguisher uses the all zeros oracle. In this case we yet again have $y_i = g_i^{r_i}$ but now $z_i = g_i^{r_i \cdot X[0,i]}$ and so we have:

$$\begin{aligned}
\prod_{i=1}^{\ell} e(y_1, \dots, y_{i-1}, z_i, y_{i+1}, \dots, y_\ell) &\equiv_p \prod_{i=1}^{\ell} e(g_1^{r_1}, \dots, g_{i-1}^{r_{i-1}}, g_i^{r_i \cdot X[0,i]}, g_{i+1}^{r_{i+1}}, \dots, g_\ell^{r_\ell}) \\
&\equiv_p \prod_{i=1}^{\ell} e(g_1^{r_1}, \dots, g_\ell^{r_\ell})^{X[0,i]} \\
&\equiv_p e(y_1, \dots, y_\ell)^{\sum_{i=1}^{\ell} X[0,i] \bmod p}.
\end{aligned}$$

But, since the group G_T is cyclic, it holds that:

$$\Pr\left[e(y_1, \dots, y_\ell)^\alpha \equiv_p e(y_1, \dots, y_\ell)^{\sum_{i=1}^{\ell} X[0,i] \bmod p}\right] = \Pr\left[\sum_{i=1}^{\ell} X[s_i, i] \equiv_p \sum_{i=1}^{\ell} X[0, i]\right] \leq 2^{-\ell} + \frac{1}{p}.$$

Hence, except with negligible probability, the distinguisher outputs 0 when given access to the all zeros oracle and we conclude that our distinguisher breaks the circular security of the scheme (with an overwhelming gap).

4 Equivalence of KDM Notions for Bit-Encryption

In this section, we establish an equivalence between the three notions of circular security for bit-encryption that were defined in Section 2.2.

Theorem 12. *For every public-key bit-encryption scheme the following are equivalent:*

1. *The scheme is circular secure with respect to key recovery.*
2. *The scheme is circular secure with respect to message recovery.*
3. *The scheme is circular secure with respect to indistinguishability of oracles.*

In particular, Theorem 12 implies that an adversary that merely distinguishes between a KDM oracle and an all zeroes oracle with a non-negligible gap can be used to fully recover the decryption-key.

Lemma 13. *Every public-key bit-encryption scheme that is circular secure with respect to key recovery is also circular secure with respect to message recovery.*

We give a sketch of the proof, for the full proof see Appendix A.

Proof Sketch. Let $(KeyGen, Enc, Dec)$ be a public-key bit-encryption scheme, and suppose that there exists an adversary A that has access to the KDM oracle and is given as input an encryption-key e and an encryption of a random bit b and manages to guess b with non-negligible advantage. We use A to construct a key-recovery adversary (which also has access to the KDM oracle).

Intuitively, it seems as though in order to find d_i (the i-th bit of the decryption-key d), the key-recovery adversary can just invoke its KDM oracle on i to obtain $c_i = Enc_e(d_i)$ and then run A on input (e, c_i) (while answering A's oracle queries using its own KDM oracle). The intuition is that since A is a message recovery attacker, it should output the bit d_i. The problem with this intuition is that A is only guaranteed to work when given an encryption of a random bit that is *independent* of the decryption-key (which is obviously not the case for d_i).

We resolve this problem by restricting our attention to the set S of all keys (e', d') for which A manages to recover messages with non-negligible advantage. We make two simple observations:

1. The set S contains a polynomial fraction of the keys (this follows from the fact that A has a non-negligible advantage over all key pairs).
2. If a fixed key pair (e, d) is in S, then there should be a non-negligible gap between the distribution $A(e, Enc_e(0))$ and the distribution $A(e, Enc_e(1))$.

Note that for a fixed (e, d), the distribution $A(e, Enc_e(d_i))$ is exactly $A(e, Enc_e(0)$ if $d_i = 0$ and $A(e, Enc_e(1))$ if $d_i = 1$. Therefore, to find d_i we approximate the following probabilities:

- The probability μ_0 that A outputs 1 when given an encryption of 0.
- The probability μ_1 that A outputs 1 when given an encryption of 1.
- The probability ν that A outputs 1 when given an encryption of d_i. (To approximate this probability we use fresh calls to the KDM oracle.)

We guess that d_i is the bit b such that ν is closer to μ_b than to μ_{1-b} and we are correct with overwhelming probability (over the coins used for the approximations). By repeating this procedure for every $i \in [|d|]$ we obtain an overwhelming probability of finding d for every $(e, d) \in S$. Since S is sufficiently large, this gives us a non-negligible probability of finding D even for a random key-pair (e, d). □

Lemma 14. *Every public-key bit-encryption scheme that is circular secure with respect to message recovery is circular secure with respect to indistinguishability of oracles.*

We give a sketch of the proof, for the full proof see Appendix A.

Proof Sketch. Let $(KeyGen, Enc, Dec)$ be a public-key bit-encryption scheme that is circular insecure with respect to indistinguishability of oracles. That is, there exists an adversary A that gets as input an encryption-key e and access to an oracle that is either the KDM oracle or the all-zeros oracle and manages to distinguish between the two cases.[16] We use A to construct a circular security message recovery adversary A' for the scheme.

For simplicity, assume that A is just given an encryption-key e and a list of ciphertexts $c_1, \ldots, c_\ell$ (where ℓ is the length of the decryption-key d) and manages to distinguish between the case that for every i the ciphertext c_i is an encryption of the i-th bit of d and the case that for every i the ciphertext c_i is an encryption of 0.[17]

We use a hybrid argument to argue that there exists an $i \in [\ell]$ such that A, given input $e, (c_1, \ldots, c_\ell)$, distinguishes between the following two cases:

1. $c_1, \ldots, c_{i-1}$ are encryptions of the first $i - 1$ bits of d and $c_i, \ldots, c_\ell$ are encryptions of 0.

2. $c_1, \ldots, c_i$ are encryptions of the first i bits of d and $c_{i+1}, \ldots, c_\ell$ are encryptions of 0.

The hybrid argument only tells us that A distinguishes the two cases for a *random* pair of keys. The first step of our message-recovery adversary A' is to find i (this can be done by approximating the output distribution of A for every hybrid with respect to random key pair) and to check that A distinguishes between the two cases for the specific keys (e, d) (where A' uses the KDM oracle to generate the two neighboring distributions).

If A does not distinguish between the two cases then A' just outputs 0 and 1 with probability $\frac{1}{2}$. If on the other hand, A does distinguish (and by the hybrid argument there is a non-negligible probability for this event), then the i-th bit of d must be 1 (otherwise the two cases are identically distributed), and therefore A' can decrypt its challenge ciphertext c by running A on $c_1, \ldots, c_{i-1}, c, c_{i+1}, \ldots, c_\ell$ where $c_1, \ldots, c_{i-1}$ are encryptions of the first $i - 1$ bits of d and $c_{i+1}, \ldots, c_\ell$ are encryptions of 0. If c is an encryption of 0 then the input to A corresponds to the $(i - 1)$-th hybrid whereas if c is an encryption of 1 then the input corresponds to the i-th hybrid. The fact that A distinguishes between these two hybrids gives A' a non-negligible advantage in guessing the value of b.

□

To complete the equivalence theorem, we also need to show the following:

[16] Actually, since Definition 6 explicitly requires semantic security, we may instead have an adversary that directly breaks semantic security. The same adversary also breaks circular security with respect to message recovery.

[17] In the general case we need to handle an adversary that can ask for t encryptions of each bit of the decryption-key, where t is a bound on the running time of the adversary. To handle this case, we construct an intermediate adversary A' that distinguishes between t encryptions of 0 and t encryptions of 1. We use an additional hybrid argument to show how to convert A' to a single message adversary (see the full proof in Appendix A for details).

Lemma 15. *Every public-key bit-encryption scheme that is circular secure with respect to indistinguishability of oracles is also circular secure with respect to key recovery.*

Intuitively, given a key recovery adversary we can obtain an indistinguishability of oracles adversary by running the key-recovery adversary using the alleged KDM oracle. If the oracle is indeed the KDM oracle then with non-negligible probability the adversary finds the decryption-key whereas if the oracle is the all zeros oracle then it should be infeasible to find the decryption-key. Since it is easy to check whether the output of the key-recovery adversary is a "good" decryption-key or not, we obtain a non-negligible advantage in distinguishing between the two oracles. See Appendix A for the full proof.

5 A Black-Box Impossibility Result

In this section we show that the bit-encryption conjecture cannot be proved by a black-box reduction. Actually, as discussed in Section 1.4, we prove a stronger result, that the circular security of every CCA-2 secure bit-encryption cannot be proved using a black-box reduction.

We start off by defining what we mean by a black-box reduction of circular security of bit-encryption to semantic security and to CCA-2 security:

Definition 16. *A* black-box reduction of circular security to semantic security for bit-encryption schemes *is a probabilistic polynomial-time algorithm R such that for every encryption scheme $(KeyGen, Enc, Dec)$ and every circular security adversary A for which there exists a polynomial p and infinitely many n such that:*

$$\left| \Pr_{(e,d)\leftarrow KeyGen(1^n)}[A^{O_{e,d}}(e)=1] - \Pr_{(e,d)\leftarrow KeyGen(1^n)}[A^{J_e}(e)=1] \right| > \frac{1}{2} + \frac{1}{p(n)}$$

there exists a polynomial p' such that for infinitely many n:

$$\Pr_{\substack{(e,d)\leftarrow KeyGen(1^n)\\ b\in_R\{0,1\}}}[R^{(KeyGen,Enc,Dec),A}(e, Enc_e(b)) = b] > \frac{1}{2} + \frac{1}{p'(n)}$$

where the probabilities are also over the coin tosses of all algorithms.

A black-box reduction of circular security to CCA-2 security is defined similarly except that the reduction R also has oracle access to the oracle Dec'_d that decrypts any message (using the decryption-key d) except for the challenge ciphertext.

We prove the following theorem:

Theorem 17. *There exists no black-box reduction of circular security to semantic security for bit-encryption schemes. Furthermore, there also exists no fully black-box reduction of circular security to CCA-2 security for bit-encryption schemes.*

Note that the furthermore clause actually implies the theorem since CCA-2 security implies semantic security. Therefore, to prove Theorem 17, it suffices to show a single encryption scheme and a successful circular security adversary for the scheme such that the scheme is CCA-2 secure even given access to the circular security adversary. Since we consider a reduction in which the circular security adversary is used in a black-box manner, we may even consider an *inefficient* circular security adversary.

For a given encryption-scheme, consider an inefficient circular security adversary A that given an encryption-key e first finds the corresponding decryption-key d (suppose that d is uniquely determined by e), then asks its oracle for encryptions of all the key bits, decrypts these ciphertexts to obtain $d'_1, \ldots, d'_n$ (where $n = |d|$) and outputs 1 if $d' \stackrel{\text{def}}{=} d'_1, \ldots, d'_n$ equals d and $\perp$ otherwise. Indeed, A breaks circular security and therefore, as stated above, to prove Theorem 17, it suffices to show a single encryption scheme for which it is infeasible to break semantic security even given oracle access to A.

Intuitively, we would like to argue that the adversary A specified above cannot be used to break the security of *any* CCA-2 secure encryption scheme (although to prove the theorem it suffices to show a single such scheme). The intuition is that for such schemes, it is infeasible, given only the encryption-key, to produce encryptions of all of the key bits.[18] Therefore, it seems as though the reduction cannot use the circular security adversary A in any meaningful way and that A can be simulated by always returning $\perp$. Thus, it seems as though the scheme remains CCA-2 secure even given oracle access to A.

The problem with the foregoing argument is that the reduction may decide to query A not on its own challenge encryption-key e but on some related key e'. In such a case we can no longer argue that A can be simulated by just returning $\perp$. While it seems strange for a *generic* reduction (which should work for any CCA-2 encryption-scheme) to run A on keys other than its own, we cannot rule out this possibility.

We overcome this difficulty by restricting our attention only to reductions that also use the encryption-scheme as a black-box. Such reductions should also work when given an *inefficient* encryption-scheme. We use this fact to construct a specific inefficient CCA-2 secure encryption scheme that has the additional important property that its encryption keys are totally unrelated. Therefore, intuitively, querying the adversary A on a key $e' \neq e$ cannot help the reduction break semantic security.

Proof of Theorem 17. We construct an *inefficient* encryption scheme $(\mathit{KeyGen}, \mathit{Enc}, \mathit{Dec})$ and an inefficient circular security adversary A for $(\mathit{KeyGen}, \mathit{Enc}, \mathit{Dec})$ such that no algorithm R that makes only polynomially many oracle calls to $(\mathit{KeyGen}, \mathit{Enc}, \mathit{Dec})$ and A can break CCA-2 security. The encryption scheme that we construct has two main properties:

1. Given only the encryption-key it is infeasible to generate encryptions of all of the bits of the private-key.

2. Encryption keys of the scheme are totally unrelated.

As is usual in black-box separations, our construction is randomized. That is, we construct a family of encryption schemes and consider a random encryption scheme in the family. Specifically, consider a totally random length tripling injective function $\mathcal{G} : \{0,1\}^n \to \{0,1\}^{3n}$ and a collection of 2^n random injective functions $\mathcal{E} \stackrel{\text{def}}{=} \{\mathcal{E}_e : \{0,1\} \times \{0,1\}^n \to \{0,1\}^{3n}\}_{e \in \mathcal{G}(\{0,1\}^n)}$. We define the following family of encryption schemes (indexed by $\mathcal{G}, \mathcal{E}$):

$\mathit{KeyGen}(1^n)$: select at random $d \in \{0,1\}^n$ and output (e, d) such that $e = \mathcal{G}(d)$.

$\mathit{Enc}_e(\sigma)$: select at random $r \in \{0,1\}^n$ and output $\mathcal{E}_e(\sigma, r)$.

$\mathit{Dec}_d(c)$: output $b \in \{0,1\}$ if there exists an $r \in \{0,1\}^n$ such that $c = \mathcal{E}_e(b, r)$, where $e = \mathcal{G}(d)$. Otherwise output $\perp$.

Note that $(\mathit{KeyGen}, \mathit{Enc}, \mathit{Dec})$ essentially form an idealized encryption scheme and that there is no correlation between different encryption keys. Additionally, note that both the set

[18] If it were feasible to generate encryptions of all the key bits than a CCA attacker could use the decryption oracle on these encryptions to find the decryption-key and break the security of the scheme.

of encryption keys and the sets of ciphertexts are a random exponentially vanishing subset of $\{0,1\}^{3n}$ and therefore a polynomially bounded adversary only has probability $\frac{\mathrm{poly}(n)}{2^{2n}} < 2^{-n}$ to produce a valid public-key or ciphertext without invoking the oracles *KeyGen* and *Enc*.

Consider the following inefficient circular security adversary for $(KeyGen, Enc, Dec)$:

$A(e,(c_1,\ldots,c_n))$: output 1 if there exist $r_1,\ldots,r_n \in \{0,1\}^n$ and $d \in \{0,1\}^n$ such that $\mathcal{G}(d) = e$ and for every $i \in [n]$, it holds that $c_i = \mathcal{E}_e(d_i, r_i)$. Otherwise output $\perp$.

The attacker A indeed breaks the circular security (with respect to indistinguishability of oracles) of $(KeyGen, Enc, Dec)$ for *every* $\mathcal{G}, \mathcal{E}$. We proceed to show that the reduction cannot utilize A to break CCA-2. That is, we will show that for every probabilistic polynomial-time algorithm R and all sufficiently large n it holds that

$$\Pr_{\substack{\mathcal{G},\mathcal{E} \\ (e,d)\leftarrow G(1^n) \\ b\in_R\{0,1\}}} [R^{(KeyGen,Enc,Dec),A,Dec'_d}(e, Enc_e(b)) = b] < \frac{1}{2} + 2\cdot 2^{-n} \tag{3}$$

where the probability is also over the coin tosses of all the algorithms and Dec'_d is the aforementioned CCA-2 decryption oracle. The existence of a single $\mathcal{G}, \mathcal{E}$ that is semantically secure follows (from standard black-box techniques, see [IR89]).

Our main step is to show that R can essentially simulate A by itself. Once we get rid of A, it is not hard to see that R cannot break semantic security.

Claim 18. *There exists a probabilistic polynomial-time algorithm R' such that for all sufficiently large n it holds that*

$$\left| \Pr_{\substack{\mathcal{G},\mathcal{E} \\ (e,d)\leftarrow G(1^n) \\ b\in_R\{0,1\}}} \left[R^{(KeyGen,Enc,Dec),A,Dec'_d}(e, Enc_e(b)) = b\right] - \Pr_{\substack{\mathcal{G},\mathcal{E} \\ (e,d)\leftarrow G(1^n) \\ b\in_R\{0,1\}}} \left[R'^{(KeyGen,Enc,Dec),Dec'_d}(e, Enc_e(b)) = b\right] \right| < 2^{-n}$$

Proof. Consider the reduction R' that on input (e,c) just runs $R(e,c)$ while monitoring R's calls to the oracles *KeyGen* and *Enc*. The new reduction R' keeps a list S_{KEYS} of all key-pairs (e',d') that R got in response to calls to *KeyGen* and a list S_{ENC} of all pairs (σ,c) of encryptions c of the bit σ that *Enc* returns with respect to e, the challenge encryption-key.

Consider an execution of R using the oracle A. There are three possible types of queries $e',(c_1,\ldots,c_n)$ made to A that R' needs to simulate:

1. There exists a d' such that $(e',d') \in S_{KEYS}$: in this case R' knows the decryption-key d' corresponding to e' and therefore can perfectly simulate the response of A by itself.

2. $e' \neq e$ and there exists no d' such that $(e',d') \in S_{KEYS}$: since the set of valid encryption keys only contains an exponentially vanishing random subset of $3n$-length strings, except with probability $\frac{\mathrm{poly}(n)}{2^{-2n}}$, it holds that e' is an invalid encryption-key and A should return $\perp$. Thus, R' simulates A by returning $\perp$.

3. $e' = e$: we separate into two cases. In the first case at least one of the ciphertexts $c_1,\ldots,c_n$ does not appear in the list S_{ENC}. Since the set of ciphertexts with respect to the key

e only contains an exponentially vanishing random subset of $3n$-bit strings, except with probability $\frac{\text{poly}(n)}{2^{2n}}$, A returns $\perp$ in this case and therefore R' just returns $\perp$.

The second case is when $c_1, \ldots, c_n$ all appear in S_{ENC}. In this case R' can retrieve from the list S_{ENC} the corresponding plaintexts $b_1, \ldots, b_n$. Now R' can simulate A by checking whether *KeyGen* returns e when given the random string $b_1, \ldots, b_n$. If so then $b_1, \ldots, b_n = d$, the decryption-key and therefore R' perfectly simulates A by returning 1. On the other hand if $b_1, \ldots, b_n \neq d$ then A can be simulated by returning $\perp$.

Since R only makes polynomially many queries, we have that the output of R' is $\frac{\text{poly}(n)}{2^{2n}} < 2^{-n}$-close to that of R. □

Thus, R' breaks the CCA-2 security of $(KeyGen, Enc, Dec)$. However, the next claim shows that *no* algorithm making polynomially many oracle queries can break the CCA-2 security of a random instance of $(KeyGen, Enc, Dec)$:

Claim 19. *For any (computationally unbounded) algorithm R' that makes at most polynomially many oracle queries and for all sufficiently large n, it holds that*

$$\Pr_{\mathcal{G},\mathcal{E},b\in_R\{0,1\}}[R'^{(KeyGen,Enc,Dec),Dec'_d}(e, Enc_e(b)) = b] < \frac{1}{2} + 2^{-n}$$

See Appendix B for the straightforward proof.

From Claims 18 and 19 we obtain Eq. (3). Using standard techniques in black-box separations (specifically an application of Markov's inequality and the Borel-Cantelli lemma, see [IR89]), the latter implies that there exist specific oracles $\mathcal{G}$ and $\mathcal{E}$ for which the corresponding encryption scheme $(KeyGen, Enc, Dec)$ is CCA-2 secure. Thus, we have found an adversary A that breaks the circular security of $(KeyGen, Enc, Dec)$ but on the other hand $(KeyGen, Enc, Dec)$ is CCA-2 secure even given oracle access to A. □

Remark. Our black-box impossibility result only considers reductions that treat both the adversary *and the primitive* (in our case the encryption scheme) as black boxes. We note that the discussion preceding the proof of Theorem 17 shows that a reduction that uses *only the adversary* as a black-box must query the adversary on keys that are somehow related to the challenge encryption-key. Since such a reduction should work for *all* bit-encryption schemes, we view this as an additional obstacle to proving the bit-encryption conjecture.

Acknowledgments

I would like to thank Benny Applebaum, Oded Goldreich and Allison Lewko for very helpful conversations and comments.

References

[ACPS09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *CRYPTO*, pages 595–618, 2009.

[App11] Benny Applebaum. Key-dependent message security: Generic amplification and completeness. In *EUROCRYPT*, pages 527–546, 2011.

[BGdMM05] Lucas Ballard, Matthew Green, Breno de Medeiros, and Fabian Monrose. Correlation-resistant storage via keyword-searchable encryption. Cryptology ePrint Archive, Report 2005/417, 2005. `http://eprint.iacr.org/`.

[BGK11] Zvika Brakerski, Shafi Goldwasser, and Yael Tauman Kalai. Black-box circular-secure encryption beyond affine functions. In *TCC*, pages 201–218, 2011.

[BHHI10] Boaz Barak, Iftach Haitner, Dennis Hofheinz, and Yuval Ishai. Bounded key-dependent message security. In *EUROCRYPT*, pages 423–444, 2010.

[BHHO08] Dan Boneh, Shai Halevi, Michael Hamburg, and Rafail Ostrovsky. Circular-secure encryption from decision diffie-hellman. In *CRYPTO*, pages 108–125, 2008.

[BRS02] John Black, Phillip Rogaway, and Thomas Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In *Selected Areas in Cryptography*, pages 62–75, 2002.

[BS03] Dan Boneh and Alice Silverberg. Applications of multilinear forms to cryptography. *Contemporary Mathematics*, 324:71–90, 2003.

[BV11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. In *FOCS*, pages 97–106, 2011.

[CL01] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *EUROCRYPT: Advances in Cryptology: Proceedings of EUROCRYPT*, 2001.

[FGHP09] Anna Lisa Ferrara, Matthew Green, Susan Hohenberger, and Michael Østergaard Pedersen. Practical short signature batch verification. In *CT-RSA*, pages 309–324, 2009.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178, 2009.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[Gol08] Oded Goldreich. *Computational complexity - a conceptual perspective*. Cambridge University Press, 2008.

[HH09] Iftach Haitner and Thomas Holenstein. On the (im)possibility of key dependent encryption. In *TCC*, pages 202–219, 2009.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28:12–24, 1999.

[IN89] Russell Impagliazzo and Moni Naor. Efficient cryptographic schemes provably as secure as subset sum. In *FOCS*, pages 236–241, 1989.

[IR89] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, STOC 1989*, pages 44–61. ACM, 1989.

[Lin06] Yehuda Lindell. A simpler construction of cca2-secure public-key encryption under general assumptions. *J. Cryptology*, 19(3):359–377, 2006.

[NY90] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC*, pages 427–437, 1990.

[RTV04] Omer Reingold, Luca Trevisan, and Salil P. Vadhan. Notions of reducibility between cryptographic primitives. In *TCC*, pages 1–20, 2004.

[Sah99] Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *FOCS*, pages 543–553, 1999.

A Full Proofs of the KDM Equivalence Lemmas

A.1 Proof of Lemma 13

Let $(KeyGen, Enc, Dec)$ be a public-key bit-encryption scheme that is circular secure with respect to key recovery. Assume toward a contradiction that there exists an efficient adversary A that breaks the circular security of $(KeyGen, Enc, Dec)$ with respect to *message recovery.* That is, there exists a polynomial $p(\cdot)$ and infinitely many $n \in \mathbb{N}$ for which:

$$\Pr_{\substack{(e,d)\leftarrow KeyGen(1^n)\\ b\in_R\{0,1\}}}\left[A^{O_{e,d}}(e, Enc_e(b)) = b\right] > \frac{1}{2} + \frac{1}{p(n)} \tag{4}$$

and therefore, using simple manipulations, we have:

$$\Pr_{(e,d)\leftarrow KeyGen(1^n)}\left[A^{O_{e,d}}(e, Enc_e(1)) = 1\right] - \Pr_{(e,d)\leftarrow KeyGen(1^n)}\left[A^{O_{e,d}}(e, Enc_e(0)) = 1\right] > \frac{1}{p(n)} \tag{5}$$

Henceforth, we only consider n from the infinite set for which Eq. (5) holds. We use A to construct an adversary A' that breaks circular security with respect to *key recovery.*

For a fixed bit $b \in \{0,1\}$, let

$$\mu_b(e,d) \stackrel{\text{def}}{=} \Pr\left[A^{O_{e,d}}(e, Enc_e(b)) = 1\right] \tag{6}$$

and let S be the following set:

$$S = \left\{(e,d)\ :\ \mu_1(e,d) - \mu_0(e,d) \geq \frac{1}{2p(n)}\right\}.$$

By Eq. (5), we have $\mathbf{E}_{e,d}\left[\mu_1(e,d) - \mu_0(e,d)\right] \geq \frac{1}{p(n)}$ and therefore, by Markov's inequality, the set S contains at least a $\frac{1}{2p(n)}$-fraction of all key-pairs.

Recall that the key-recovery adversary A' is given an encryption-key e and access to the KDM oracle $O_{e,d}$ and needs to find d. We show how to recover d bit-by-bit but only in the case that $(e,d) \in S$. Considering only keys that are in S suffices because S contains a polynomial fraction of all key-pairs, and therefore succeeding only for keys in S gives us a non-negligible probability to succeed in general.

Let (e,d) be a fixed key-pair in S. We proceed to describe an efficient procedure that given e and oracle access to $O_{e,d}$, finds d. To do so, we begin by estimating $\mu_0(e,d)$ and $\mu_1(e,d)$. This can be done by taking the average of $O(p(n)^2 n)$ independent executions of A (each execution is done with fresh randomness and fresh oracle responses). Since the key is fixed, each invocation is totally independent and by the Chernoff bound, with probability at least $1-2^{-n}$, our approximation $\hat{\mu}_b(e,d)$ will be $\frac{1}{16p(n)}$-close to $\mu_b(e,d)$.

Once this initial preprocessing step is complete, we can recover d bit-by-bit. To recover the i-th bit of d, the algorithm A' attempts to approximate

$$\nu_i(e,d) \stackrel{\text{def}}{=} \Pr\left[A^{O_{e,d}}(e, Enc_e(d_i)) = 1\right]. \tag{7}$$

As before, this is done by invoking A with fresh oracle queries and with fresh encryptions of d_i that are obtained by querying the oracle $O_{e,d}(i)$. Invoking A independently $O(p(n)^2 n)$ times gives us an approximation $\hat{\nu}_i(e,d)$ that is $\frac{1}{16p(n)}$-close to $\nu_i(e,d)$ (with probability at least $1-2^{-n}$).

Claim 20. *For every $(e,d) \in S$, except with exponentially vanishing probability, $\hat{\nu}_i(e,d)$ is closer to $\hat{\mu}_{d_i}(e,d)$ than to $\hat{\mu}_{1-d_i}(e,d)$.*

The claim follows from the following straightforward calculation.

Proof. Clearly $\nu_i(e,d) = \mu_{d_i}(e,d)$ and as argued above (except with exponentially vanishing probability) $|\hat{\nu}_i(e,d) - \nu_i(e,d)| < \frac{1}{16p(n)}$ and $|\hat{\mu}_b(e,d) - \mu_b(e,d)| < \frac{1}{16p(n)}$. Thus, we have:

$$\begin{aligned} |\hat{\nu}_i(e,d) - \hat{\mu}_{d_i}(e,d)| &\le |\hat{\nu}_i(e,d) - \nu_i(e,d)| + |\nu_i(e,d) - \mu_{d_i}(e,d)| + |\mu_{d_i}(e,d) - \hat{\mu}_{d_i}(e,d)| \\ &< \frac{1}{8p(n)}. \end{aligned}$$

And on the other hand we have:

$$\begin{aligned} |\hat{\nu}_i(e,d) - \hat{\mu}_{1-d_i}(e,d)| &\ge |\hat{\mu}_{d_i}(e,d) - \hat{\mu}_{1-d_i}(e,d)| - |\hat{\nu}_i(e,d) - \hat{\mu}_{d_i}(e,d)| \\ &> |\hat{\mu}_{d_i}(e,d) - \mu_{d_i}(e,d) + \mu_{d_i}(e,d) - \mu_{1-d_i}(e,d) + \mu_{1-d_i}(e,d) - \hat{\mu}_{1-d_i}(e,d)| - \frac{1}{8p(n)} \\ &\ge |\mu_{d_i}(e,d) - \mu_{1-d_i}(e,d)| - |\hat{\mu}_{d_i}(e,d) - \mu_{d_i}(e,d)| - |\mu_{1-d_i}(e,d) - \hat{\mu}_{1-d_i}(e,d)| - \frac{1}{8p(n)} \\ &\ge \frac{1}{2p(n)} - \frac{1}{16p(n)} - \frac{1}{16p(n)} - \frac{1}{8p(n)} \\ &= \frac{1}{4p(n)} \end{aligned}$$

where the last inequality follows from the fact that $|\mu_{d_i} - \mu_{1-d_i}| \ge \frac{1}{2p(n)}$ (since $(e,d) \in S$). □

Therefore, we guess that the i-th bit of d is the bit b such that ν_i is closer to μ_b than to μ_{1-b} and by Claim 20 we are correct (for a particular i) except with exponentially vanishing probability. Taking a union bound over all i we find all of d except with exponentially vanishing probability.

We showed that if $(e,d) \in S$, we find d with overwhelming probability and since S contains a polynomial fraction of all valid keys, in the general case, we find d with non-negligible probability.

A.2 Proof of Lemma 14

Let $(KeyGen, Enc, Dec)$ be a public-key bit-encryption scheme that is circular secure with respect to message recovery. Assume toward a contradiction that there exists an efficient adversary A that breaks the circular security of $(KeyGen, Enc, Dec)$ with respect to indistinguishability of oracles. As remarked in Footnote 16, it may be the case that A directly breaks semantic security, however this directly contradicts our circular security with respect to message recovery assumption. Otherwise, there exists a polynomial $p(\cdot)$ and infinitely many $n \in \mathbb{N}$ for which:

$$\left| \Pr_{(e,d) \leftarrow KeyGen(1^n)} \left[A^{O_{e,d}}(e) = 1 \right] - \Pr_{(e,d) \leftarrow KeyGen(1^n)} \left[A^{J_e}(e) = 1 \right] \right| \geq \frac{1}{p(n)} \tag{8}$$

and without loss of generality[19] we have:

$$\Pr_{(e,d) \leftarrow KeyGen(1^n)} \left[A^{O_{e,d}}(e) = 1 \right] - \Pr_{(e,d) \leftarrow KeyGen(1^n)} \left[A^{J_e}(e) = 1 \right] \geq \frac{1}{p(n)}. \tag{9}$$

Henceforth, we only consider $n \in \mathbb{N}$ from the infinite set for which Eq. (9) holds.

Let $t(n)$ be a polynomial bounding the number of queries that A makes to its KDM oracle for security parameter 1^n. We use A to construct an adversary A' that gets as input $t(n)$ ciphertexts and using access to the KDM oracle, distinguishes between the case that the ciphertexts are all encryptions of 0 and the case that they are all encryptions of 1. Using a standard hybrid argument A' can be transformed into a standard message-recovery adversary.

For simplicity, we assume (without loss of generality) that for every security parameter n, all decryption keys have the same (polynomially-bounded) length $\ell \stackrel{\text{def}}{=} \ell(n)$. For $0 \leq i \leq \ell$, let $H_{e,d}^{(i)}$ be the following hybrid oracle:

$$H_{e,d}^{(i)}(j) = \begin{cases} Enc_e(0) & \text{if } j \leq i \\ Enc_e(d_j) & \text{otherwise} \end{cases}.$$

Observe that $H_{e,d}^{(0)} \equiv O_{e,d}$ and that $H_{(e,d)}^{(\ell)} \equiv J_e$. For $i \in \{0, \ldots, \ell\}$ let:

$$\mu_i(e,d) = \Pr\left[A^{H_{e,d}^{(i)}}(e) = 1 \right]. \tag{10}$$

By Eq. (9) we have $\mathbf{E}_{e,d}\left[\mu_0(e,d) - \mu_\ell(e,d)\right] \geq \frac{1}{p(n)}$. Therefore, by a hybrid argument, there exists $0 \leq i^* \leq \ell - 1$ such that:

$$\mathop{\mathbf{E}}_{e,d}\left[\mu_{i^*}(e,d) - \mu_{i^*+1}(e,d)\right] \geq \frac{1}{p(n)\ell}$$

We will attempt to find i^*. To do so, note that by the Chernoff bound, except with negligible probability, for every i we can find a $\frac{1}{4p(n)\ell}$-approximation of $\mathbf{E}_{e,d}[\mu_i(e,d)]$ by repeating the following process $O((p(n)\ell)^2 n)$ times and taking the average:

1. Select random $(e', d') \leftarrow G(1^n)$.
2. Output $A^{H_{e',d'}^{(i)}}(e')$ (note that we can implement the oracle since we have d').

Therefore, except with negligible probability, we can efficiently find an i^* such that:

$$\mathop{\mathbf{E}}_{e,d}\left[\mu_{i^*}(e,d) - \mu_{i^*+1}(e,d)\right] \geq \frac{1}{2p(n)\ell} \tag{11}$$

[19] We do the standard trick: if Eq. (8) holds for infinitely n then either Eq. (9) holds for infinitely many n as-is or it holds if we complement A's output.

Recall that our goal is to construct an adversary A' that is given an encryption-key e and ciphertexts $c_1, \ldots, c_{t(n)}$ as well as oracle access to $O_{e,d}$ and needs to decide whether the ciphertexts are all encryptions of 0 or all encryptions of 1. The second step of A' (after finding i^*) is to test if the gap specified by Eq. (11), which holds for a random key-pair, also holds in practice for the specific keys given as input.

Fix $\epsilon = \frac{1}{8p(n)\ell}$ and let $S \stackrel{\text{def}}{=} \{(e,d) \leftarrow G(1^n)\colon \mu_{i^*}(e,d) - \mu_{i^*+1}(e,d) \geq \epsilon\}$. Note that by Eq. (11) and Markov's inequality, the set S contains at least an ϵ fraction of keys. Our next step is to attempt to ascertain whether $(e,d) \in S$. To do so we approximate the values of $\mu_{i^*}(e,d)$ and $\mu_{i^*+1}(e,d)$. This approximation can be done by taking the average output of $A(e)$ over $O(\frac{1}{\epsilon}^2 n)$ trials while answering A's oracle calls as necessary (by either returning $Enc_e(0)$ or using the KDM oracle that A' has access to) giving us an $\frac{\epsilon}{8}$ approximation $\hat{\mu}_{i^*}(e,d)$ and $\hat{\mu}_{i^*+1}(e,d)$ of $\mu_{i^*}(e,d)$ and $\mu_{i^*+1}(e,d)$ (respectively).

If the gap between $\hat{\mu}_{i^*}(e,d)$ and $\hat{\mu}_{i^*+1}(e,d)$ is less than $\frac{\epsilon}{4}$ then A' outputs 0 and 1 with probability $\frac{1}{2}$. However, except with negligible probability, this only happens if $(e,d) \notin S$. Therefore, there is non-negligible probability that the gap between $\hat{\mu}_{i^*}(e,d)$ and $\hat{\mu}_{i^*+1}(e,d)$ is larger than $\frac{\epsilon}{4}$.

If we are in the case that the gap between $\hat{\mu}_{i^*}(e,d)$ and $\hat{\mu}_{i^*+1}(e,d)$ is in fact larger than $\frac{\epsilon}{4}$, then (except with negligible probability) for the fixed keys (e,d) it holds that:

$$\mu_{i^*}(e,d) - \mu_{i^*+1}(e,d) > \frac{\epsilon}{2}. \tag{12}$$

Notice that this gap also implies that the i^*-th bit of d must be 1 (since d is fixed, having $d_{i^*} = 0$ would imply that the two experiments are identical).

We will show that Eq. (12) allows A' to distinguish between $c_1, \ldots, c_{t(n)}$ which are all either encryptions of 0 or all encryptions of 1 (recall that $t(n)$ is a polynomial bounding the number of oracle queries that A makes). To do so, A' invokes $A(e)$ while answering A's oracle queries for bit location $1 \leq j \leq \ell$ of the decryption-key as follows:

- If $j < i^*$ then A' just answers with $Enc_e(0)$.
- If $j > i^*$ then A' queries its own KDM oracle $O_{e,d}(j)$ and returns its answer.
- If $j = i^*$ and this is the k-th query, then A' returns c_k.

To complete the proof, notice that if $c_1, \ldots, c_{t(n)}$ are encryptions of 0 then the input to A is precisely the i^*-th hybrid and therefore A' outputs 1 with probability $\mu_{i^*}(e,d)$. On the other hand, if $c_1, \ldots, c_{t(n)}$ are encryptions of 1 then the input to A is the (i^*+1)-th hybrid (by our observation that $d_{i^*} = 1$). By Eq. (12) there is a non-negligible gap between A's output distribution in these two cases and therefore A' manages to distinguish between $t(n)$ encryptions of 0 and $t(n)$ encryption of 1 with non-negligible probability. Using a standard hybrid argument, A' can be transformed into an adversary that distinguishes between a *single* encryption of either 0 or 1 which in turn can be easily transformed into a message-recovery adversary.

A.3 Proof of Lemma 15

Let $(KeyGen, Enc, Dec)$ be a public-key bit-encryption scheme that is circular secure with respect to indistinguishability of oracles. Assume toward a contradiction that there exists an efficient adversary A that breaks the circular security of $(KeyGen, Enc, Dec)$ with respect to key recovery.

We use A to construct an adversary A' that on input an encryption-key e, distinguishes between a KDM oracle and the all-zeros oracle. The adversary A', on input an encryption-key e, works as follows:

1. Invoke $A(e)$ to obtain a potential decryption-key d' while answering A's oracle queries with A''s oracle (which is either the KDM oracle or the all-zeros oracle).

2. For $i = 1, \ldots, n$:

   (a) Choose a random $b \in_R \{0,1\}$.

   (b) If $Dec_{d'}(Enc_e(b)) \neq b$ output 0 and halt.

3. Output 1.

If the oracle is the KDM oracle $O_{e,d}$, then with non-negligible probability A outputs the correct decryption-key d and then A' outputs 1. On the other hand, if the oracle is the all-zeros oracle J_e then A' outputs 0 except with negligible probability. To prove the latter we first need to show the following claim:

Claim 21.

$$\Pr_{\substack{(e,d) \leftarrow KeyGen(1^n) \\ d' \leftarrow A^{J_e}(e)}} \left[\Pr_{b \in_R \{0,1\}} [Dec_{d'}(Enc_e(b)) = b] > \frac{2}{3} \right] < \mathrm{neg}(n)$$

Proof. We say that a key pair (e, d') is good if $\Pr_{b \in_R \{0,1\}} [Dec_{d'}(Enc_e(b)) = b] > \frac{2}{3}$. To prove the claim, consider a semantic security adversary for the encryption scheme that on input e runs $A(e)$ while answering A's queries with fresh encryptions of 0 to obtain d'. Then, the semantic security adversary approximates whether the key pair (e, d') is a good key pair. If so then then the adversary uses d' to decrypt the challenge ciphertext and if (e, d') is not a good key pair then the adversary outputs 0 and 1 with probability $\frac{1}{2}$.

If the claim is false then the probability that (e, d') are good is non-negligible and the proposed adversary breaks the semantic security of the scheme. □

Therefore, if the oracle is the all-zeros oracle, then except with negligible probability, in each iteration the adversary has a $\frac{2}{3}$ probability to halt and output 0 in step (2b) (which implies an exponentially vanishing probability to output 1). Hence, A' distinguishes between the KDM oracle and the all-zeros oracle with a non-negligible gap.

B Proof of Claim 19

To proof the claim, we consider first the following simplification. Let $f : \{0,1\} \times \{0,1\}^n \rightarrow \{0,1\}^{3n}$ be a random injective function. We will show that for every algorithm A that makes at most polynomially many oracle queries:

$$\Pr_{f, b \in_R \{0,1\}, r \in_R \{0,1\}^n} [A^{f,g}(f(b,r)) = b] < \frac{1}{2} + 2^{-n}$$

where g inverts f in the following sense: on input $c \in \{0,1\}^{3n}$ the function g outputs $b' \in \{0,1\}$ if there exists an $r' \in \{0,1\}^n$ such that $c = (b', r')$ and $(b', r') \neq (b, r)$ and otherwise outputs $\perp$.

To prove the simplified case, notice that since f is chosen uniformly at random, with probability $\frac{\mathrm{poly}(n)}{2^{2n}}$ the adversary A never queries the oracles f or g on r. In this case, conditioned on the view of A, the value of b is an independent random coin flip which can only be guessed with probability $\frac{1}{2}$. Thus,

$$\Pr_{f, b \in_R \{0,1\}, r \in_R \{0,1\}^n} [A^{f,g}(f(b,r)) = b] \leq \frac{\mathrm{poly}(n)}{2^{2n}} \cdot 1 + \left(1 - \frac{\mathrm{poly}(n)}{2^{2n}}\right) \cdot \frac{1}{2} < \frac{1}{2} + 2^{-n}.$$

The general claim actually follows by a reduction to the simplified case. Specifically, an adversary for the general claim can be used to obtain an adversary for the simplified case by answering oracle queries of the form $Enc_e(b; r)$ with $f(b; r)$ and queries of the form $Dec'_d(c)$ with $g(c)$. If the adversary successfully guesses b in the general case then we get an adversary for the simplified case.