Bounded Key-Dependent Message Security

Boaz Barak* Iftach Haitner† Dennis Hofheinz‡ Yuval Ishai§

March 8, 2010

Abstract

We construct the first public-key encryption scheme that is proven secure (in the standard model, under standard assumptions) even when the attacker gets access to encryptions of arbitrary efficient functions of the secret key. Specifically, under either the DDH or LWE assumption, for all polynomials L and N we obtain a public-key encryption scheme that resists key-dependent message (KDM) attacks for up to $N(k)$ public keys and functions of *circuit size* up to $L(k)$, where k denotes the size of the secret key. We call such a scheme *bounded KDM secure*. Moreover, we show that our scheme suffices for one of the important applications of KDM security: the ability to securely instantiate symbolic protocols with axiomatic proofs of security.

We also observe that any fully homomorphic encryption scheme which additionally enjoys circular security and circuit privacy is *fully KDM secure* in the sense that the encryption and decryption algorithms can be independent of the polynomials L and N as above. Thus, the recent fully homomorphic encryption scheme of Gentry (STOC 2009) is fully KDM secure under certain non-standard hardness assumptions.

Previous works obtained either full KDM security in the random oracle model (Black et al., SAC 2002) or security with respect to a very restricted class of functions (e.g., clique/circular security and affine functions, Boneh et al., CRYPTO 2008, and Applebaum et al., CRYPTO 2009).

Our main result is based on a combination of the circular-secure encryption scheme of either Boneh et al. or Applebaum et al. with Yao's garbled circuit construction.

Finally, we extend the impossibility result of Haitner and Holenstein (TCC 2009), showing that it is impossible to prove KDM security against a family of query functions that contains exponentially hard pseudorandom functions, using only *black-box* access to the query function and the adversary attacking the scheme. This proves that the non-black-box usage of the query function in our proof of security makes to the KDM query function is *inherent*.

Keywords: KDM/clique/circular security; fully homomorphic encryption; formal security.

*Princeton University, boaz@cs.princeton.edu . Supported by NSF grants CNS-0627526, CCF-0426582 and CCF-0832797, US-Israel BSF grant 2004288 and Packard and Sloan fellowships.

†Microsoft Research New England, iftach@microsoft.com

‡Karlsruhe Institute of Technology, Dennis.Hofheinz@kit.edu Part of work performed while at CWI and supported by an NWO Veni grant.

§Technion and UCLA, yuvali@cs.technion.ac.il

1 Introduction

An encryption scheme is *key-dependent message (KDM) secure* if it is secure even against an attacker who has access to encryptions of messages that depend on the secret key. This strong notion of security, introduced by Black et al. [6], tries to capture scenarios where there could be correlations between the secret key and the encrypted messages. At a first glance, it may seem that such correlations only arise from bugs or errors on part of the protocol designer, and hence achieving such a strong security is not of much importance. It turns out, however, that such attacks naturally occur when considering complex systems. For example, in some popular disk encryption utilities, the disk encryption key can end up being stored in the page file, and thus is encrypted along with the disk content [7]. In addition, Camenisch and Lysyanskaya [9] showed that schemes with a certain restricted form of KDM security known as "circular security" are useful for constructing *Anonymous Credential Systems*. Finally, and perhaps most importantly, KDM security naturally arises as the right notion when one wishes to securely instantiate symbolic protocols with an axiomatic proof of formal security (see Section 6).

For a while, building a KDM-secure encryption scheme in the standard model, under any well studied hardness assumption, seemed too hard a nut to crack. The only scheme that was shown to resist any kind of KDM attacks was given by Black et al. [6] in the random-oracle model. Yet, in recent years KDM-secure encryption schemes were given for some non-trivial families of functions. This line of work started with the works of Halevi and Krawczyk [19] and Hofheinz and Unruh [20], who gave private-key encryption schemes secure against significantly restricted classes of KDM queries. Concretely, [19] prove security against arbitrary but fixed KDM queries that are known in advance, and against KDM queries that do not depend on certain "protected" parts of the key. The constructions from [20] obtain statistical KDM security in the presence of sufficiently few (arbitrary) KDM queries, as well as a stateful KDM-secure scheme in which KDM queries may only depend on the current state (but not on previous states).

A major step was taken by Boneh, Halevi, Hamburg, and Ostrovsky [7] who presented, under the *decisional Diffie-Hellman (DDH)* assumption, a public-key encryption scheme that is $N(k)$-circular secure for every polynomial N, and in fact is secure against the more general family of attacks allowing the adversary access to encryptions of arbitrary affine functions of the vector of $N(k)$ secret keys. Applebaum, Cash, Peikert, and Sahai [4] presented more efficient schemes that are secure against a similar family of key-dependent attacks, whose security is based on different assumptions: the *learning parity with noise (LPN)* assumption in the secret-key case and the *learning with errors (LWE)* assumption in the public-key case. In a recent independent work, Brakerski, Goldwasser, and Kalai [8] presented a transformation from a KDM secure scheme satisfying a certain property (in particular satisfied by the DDH and LWE based schemes of [7, 4]) into a scheme that is KDM secure with respect to a larger class of functions. While their transformation cannot be used to achieve security against all circuits of size $p(n)$, it has the benefit of depending only on the number of functions in the class, and being independent of their circuit size or number of keys. In particular they achieve KDM security with respect to the class of constant degree polynomials and any polynomial number of keys.

Despite the above progress, the families of functions for which KDM security was achieved prior to our work (in the standard model, under standard assumptions) was still quite restricted. In particular, these families were not sufficiently rich for several of the applications of KDM security in the context of complex systems and formal protocols. A partial explanation for this rather limited

success was recently given by Haitner and Holenstein [18], who showed the impossibility of obtaining KDM security based on standard assumptions and using standard techniques. (In Section 1.2, we will describe their results in more detail, since we will later extend them to our case of bounded KDM security.)

1.1 Our Results

Our main result is the following:

Theorem 1.1 (Informal)**.** *Under the DDH or LWE assumption, for any given polynomials $L = L(k)$ and $N = N(k)$, there exists a public-key encryption scheme that is KDM-secure with respect to the class of circuits of size $L(k)$, and for $N(k)$ independent keys, where k denotes the size of the keys.*

We call such a scheme a *bounded KDM-secure* encryption scheme. (This is in contrast with a *fully-* or *unbounded*-KDM scheme, where the circuit size and the number of keys can be an arbitrarily large polynomial in the security parameter, independent of the scheme's complexity.) We argue that this is the first encryption scheme (under standard cryptographic assumptions) that handles a rich enough function class to capture most "real life" KDM attacks.

The original motivation for KDM security was to securely instantiate symbolic cryptographic protocols that have a formal proof of security in some axiomatic system. As further evidence for the usefulness of bounded KDM security, we show that our notion is strong enough for this application:

Theorem 1.2 (Informal)**.** *Let P be a symbolic protocol with operations such as public-key encryption and digital signatures. Then, instantiating P with a bounded KDM-secure[1] encryption scheme provides a computationally sound implementation.*

This yields the first soundness result without restrictions (such as assuming protocols without key-cyclic expressions) in the standard model.

Finally, we show that the above positive results are tight, by extending an impossibility result of Haitner and Holenstein [18] in the following sense:

Theorem 1.3 (Informal)**.** *An encryption scheme cannot be proven to be KDM-secure against a family of functions that contains exponentially hard pseudorandom functions, if the proof of security only accesses the query function and the adversary attacking the scheme in a black-box manner (i.e., as oracles).*

Remarks. We note the following points about our result:

1. *Efficiency.* Our scheme, although polynomial time, is not practically efficient as it uses the garbled circuit construction and its ciphertext length is at least L, where L is a bound on the circuit size of the KDM function. There are more efficient candidate KDM-secure cryptosystems if one is willing to settle for non-standard assumptions or the random oracle model.

2. *Full KDM security.* Although we only prove our scheme to be bounded KDM secure, it is of course possible that it is KDM secure with respect to *any* efficient KDM function. In fact,

[1] Actually, the precise notion we use is *length-dependent* KDM security (see Definition 6.1). This is a slight strengthening of bounded KDM security, and our scheme satisfies this stronger notion as well.

there seems to be an interesting obstacle to any KDM attack on our scheme. Suppose that we instantiate the scheme to be secure with respect to KDM functions of size k^3. Now suppose that there is a successful KDM attack against it, and for simplicity assume the attack consists of getting one encryption of $h(sk)$ where h is some efficiently computable function. Then the success of this attack implies that either DDH is false (assuming we instantiate our scheme from the DDH assumption), or that h has no circuit of size k^3. Hence, a proof that this construction is insecure against a polynomial-time KDM attacker will provably demonstrate than either DDH is false, or that $\mathbf{P} \not\subseteq \mathbf{Size}(k^2)$ (we lose a factor of k because h has a k-bit output). The latter is a widely believed fact, but its proof would be considered a major breakthrough in complexity theory. (Also, it is not at all clear how to derive such a conclusion directly from the DDH assumption— typically in cryptography we need to use *subexponential* hardness assumptions to get such a condition.) More generally, a successful attack is some way to *certify* that h is hard— even though it is easy in time k^3 to generate a random function outside of $\mathbf{Size}(k^2)$, it is not at all clear how to generate such a function along with a publicly verifiable certificate of hardness.

3. *Black-box-ness.* Our scheme makes a non-black-box use of the KDM function h, where Theorem 1.3 shows that this use is *inherent*.

1.1.1 Applications to formal security.

A central motivation for the study of KDM security lies in the connection between formal and computational cryptography. In formal cryptography (starting with [13, 14, 25]), cryptographic operations like encryption or digital signatures are abstracted as symbolic operators that (only) adhere to natural rules. Given such rules, a simple calculus enables machine-assisted security analysis.

It was proven by Adão et al. [2] that *fully* KDM-secure encryption schemes imply computational soundness for *arbitrary* symbolic protocols. We reconsider their proof and show (Theorem 6.2) that bounded KDM security of the type that we achieve suffices. Hence, our combined results give the first encryption scheme (under standard cryptographic assumptions) whose security implication can be verified using formal security methods.

We stress that the clique security achieved by [7, 4] only enables to apply these formal methods to a very limited class of applications. For more details see Section 6.

1.2 Our Techniques

We now give an informal overview of the proof of Theorem 1.1. The following exposition focuses on a scheme that is secure against a *single-key* KDM attack. That is, there is only one public/private key pair (pk, sk) of length k, and the attacker can obtain encryptions of messages of the form $h(sk)$ for an arbitrary function h of circuit complexity at most $L(k)$. (Here $L = L(k)$ is an arbitrary fixed polynomial which affects the complexity of the encryption and decryption, but not the complexity of key generation.) The multiple-key case raises some additional subtleties that we ignore for the moment.

Recall that a *homomorphic* public-key encryption scheme is a public-key encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ that also has an additional algorithm Eval for evaluating functions on an encrypted message. Concretely, Eval takes the public key pk, an encryption of a message M, and a description

of a function h from some family $\mathcal{H}$, and outputs a string from which $h(M)$ can be efficiently decrypted using the secret key sk. Our starting point is the following observation: a sufficiently strong homomorphic encryption is in fact also KDM-secure (with respect to the same class of functions $\mathcal{H}$), where "sufficiently strong" means that the scheme satisfies the following additional properties:

1. Self-referential (i.e., 1-circular) security: $\mathsf{Enc}_{pk}(sk)$ is indistinguishable from $\mathsf{Enc}_{pk}(0^k)$.
2. Strong function privacy: For every $h \in \mathcal{H}$ and plaintext M, $\mathsf{Eval}_{pk}(h, \mathsf{Enc}_{pk}(M))$ is indistinguishable from $\mathsf{Enc}_{pk}(h(M))$, even against a distinguisher that knows the secret key.

The basic idea for proving the KDM-security of such a scheme is that a distinguisher between $\mathsf{Enc}_{pk}(h(sk))$ and $\mathsf{Enc}_{pk}(h(0^k))$ can be used to distinguish between $\mathsf{Enc}_{pk}(sk)$ and $\mathsf{Enc}_{pk}(0^k)$ by simply running Eval with the function h (and thus the "KDM queries" are useless). When turning this idea into a proof one sees that it is crucial that function privacy hold even with respect to a distinguisher that knows the secret key.

This observation already implies that Gentry's recent breakthrough fully homomorphic encryption scheme [16] is fully KDM-secure, assuming that it is circular-secure (an assumption which is anyway necessary in Gentry's case to get a truly fully homomorphic encryption, where the public key does not grow with the depth of the circuit).[2] Since all *natural* candidates for public-key encryption schemes are not known to be 1-circular insecure, we find this observation interesting, as the assumption of circular security seems cleaner and more conservative than assuming full KDM security. (In particular, it is more easily "falsifiable" in the sense of Naor [27].)

In fact, it turns out that it suffices to have only *weak* function privacy, requiring that $\mathsf{Eval}_{pk}(h, \mathsf{Enc}_{pk}(M))$ be indistinguishable from $\mathsf{Eval}_{pk}(h', \mathsf{Enc}_{pk}(M))$ for h, h', M such that $h(M) = h'(M)$ (again indistinguishability is with respect to attackers who know the secret key).[3] See Theorem 2.4 for the details.

The latter observation suggests an approach to get KDM security for circuits of size L under standard assumptions. Consider any two-message protocol for evaluating a *universal function* with security against semi-honest parties. Such a protocol takes an input M from a receiver and a circuit h from a sender, and delivers the output $h(M)$ to the receiver. Given any such protocol and a standard public-key encryption (PKE), one can construct a homomorphic scheme with *weak* function privacy as follows. The public key is the public key pk of the PKE. The encryption of M under pk is a triple (C, pk', C') where C is an encryption of M under pk, pk' is the receiver's first message in the protocol on input M, and C' is an encryption under pk of the secret randomness sk' used to generate pk' (which is needed to recover the output of the protocol). The algorithm $\mathsf{Eval}((C, pk', C'), h)$ returns the sender's response to pk' on input h along with C'. Given sk, the output of Eval can be used to decrypt $h(M)$ by first recovering sk' and then computing the receiver's output in the protocol.

The advantage of this approach is that it can be instantiated under standard assumptions by using Yao's protocol [30]. More concretely, a secure two-message protocol for the universal function can be obtained by combining Yao's garbled circuit construction and any *two-message oblivious*

[2] As Gentry notes, if one assumes his scheme is circular-secure then it also enjoys strong statistical function privacy.

[3] This is indeed a weaker notion since if $y = h(M) = h'(M)$ then one can see that strong function privacy implies that $\mathsf{Eval}(h, \mathsf{Enc}(M)) \approx \mathsf{Enc}(y) \approx \mathsf{Eval}(h', \mathsf{Enc}(M))$. Intuitively, weak function privacy allows Eval to map ciphertexts from one domain to a different domain, while this is ruled out by strong function privacy.

transfer (OT) [29, 15] protocol.[4] Unlike the alternative of using *fully* homomorphic encryption, however, this protocol has the caveat that the communication must grow with the size of h, and hence (weak) function privacy can only hold with respect to the class $\mathcal{H}$ of all circuits with some a-priori size bound L.

A more subtle problem is that of making the homomorphic scheme constructed in the above way circular-secure. Indeed, encrypting the secret key of the homomorphic scheme with its own public key results in a circular dependency between the underlying PKE and the two-message protocol: the secret key sk of the PKE is encrypted using the "public key" pk' of the protocol, whereas the "secret key" sk' of the protocol is encrypted using the public key pk of the PKE. Even if the PKE is circular-secure, it is not clear that this property will be respected by the above construction.

Our way to handle this difficulty is by introducing a new notion that we call "targeted encryption", which is aimed towards resolving the above dependency when applied to a two-message protocol based on Yao's technique. Targeted encryption can be viewed as a circular-secure extension of both public-key encryption and two-message OT. Loosely speaking, one can think of this as an OT protocol where the receiver has *no secret information* apart from the input selection vector s. This may look strange at first, and indeed it can be shown to be inherently at odds with the standard notion of OT, which requires that the sender learn nothing about s. But it turns out that one can obtain a meaningful relaxation of the above notion that is strong enough for our purposes. We then show that both the schemes of Boneh et al. [7] and Applebaum et al. [4] can be used to construct targeted encryption. The key property we use is that both schemes enjoy KDM security against *affine functions*, and in fact this is proven by giving a public algorithm to compute an encryption of any affine function of the secret key. We show that such an algorithm implies targeted encryption.

As mentioned above, multiple-key security adds some additional difficulties. In particular, targeted encryption on its own does not seem sufficient for *multiple key* security, and to handle this case we need to appeal to the multiple-key circular security of the underlying schemes.

To show our negative result (Theorem 1.3), we employ the techniques of Haitner and Holenstein [18]. Concretely, they showed that an encryption scheme cannot be proved to be KDM-secure against the family of *all* functions, if the proof of security only accesses the query function and the adversary attacking the scheme in a black-box manner (i.e., as oracles).[5] Here we extend this result to *every* family of functions that contains exponentially hard pseudorandom functions. There was no prior scheme that was shown (under a standard assumption) to be secure with respect to such a family, although many of the applications of KDM security require that the KDM function can be a cryptographic primitive such as a signature, a hash function, etc.

2 Preliminaries

Notation. For $n \in \mathbb{N}$, let $[n] := \{1, \ldots, n\}$. Throughout the paper, $k \in \mathbb{N}$ denotes the security parameter. For a finite set X, we denote by $x \leftarrow X$ the process of sampling x uniformly from

[4] Two-message OT is a protocol comprised of one message from the receiver and one message from the sender, where the receiver has an input selection bit s and the sender has a pair of input strings X_0, X_1. In the end of the protocol the receiver only learns X_s and the sender learns nothing about s. Here we need k parallel instances of OT, where the receiver has a k-bit input selection vector s and the sender has k pairs of strings.

[5] They also showed that it is impossible to prove (in a black-box way) that a trapdoor-permutation based scheme is KDM-secure against a family of t-wise independent hash functions, for t that is longer than the ciphertext size (here a non-black-box access to the query function is allowed).

X. For a probabilistic algorithm A, we denote by $y \leftarrow A(x)$ the process of running A on input x and with uniform randomness, and assigning y the result. If A runs in time polynomial in the security parameter k, then A is a PPT machine. (We always assume that k can be efficiently computed from the input to the algorithm even if it not explicitly given.) A function $f : \mathbb{N} \to [0,1]$ is *negligible* iff $\forall c \in \mathbb{N}\, \exists k_0 \in \mathbb{N}\, \forall k > k_0 : |f(k)| < k^{-c}$. We say f is *overwhelming* iff $1 - f$ is negligible. Two collections $X = (X_k)_{k\in\mathbb{N}}$ and $Y = (Y_k)_{k\in\mathbb{N}}$ of random variables are *computationally indistinguishable*, written $X \overset{c}{\approx} Y$, iff for every nonuniform polynomial-time distinguisher D, we have that $\mathsf{Pr}\left[D(1^k, X_k) = 1\right] - \mathsf{Pr}\left[D(1^k, Y_k) = 1\right]$ is negligible. We use $\circ$ for concatenation.

Encryption schemes. A public-key encryption (PKE) scheme with message space $\mathcal{M} = \mathcal{M}_k$ and secret key space $\mathcal{K} = \mathcal{K}_k$, consists of three algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ — Key generation $\mathsf{Gen}(1^k)$ outputs a public key pk and a secret key $sk \in \mathcal{K}_k$. Encryption $\mathsf{Enc}_{pk}(M)$ takes a public key pk and a message $M \in \mathcal{M}_k$, and outputs a ciphertext C. Decryption $\mathsf{Dec}_{sk}(C)$ takes a secret key sk and a ciphertext C, and outputs a message M. For correctness, we require $\mathsf{Dec}_{sk}(C) = M$ for all $M \in \mathcal{M}_k$, all (pk, sk) in the range of $\mathsf{Gen}(1^k)$, and all C in the range of $\mathsf{Enc}_{pk}(M)$. For simplicity, we will assume from now on that both the key space and the message space are $\{0,1\}^k$. Our definitions and results, however, can be easily adapted to the case of messages of arbitrary length.

2.1 Garbled circuits

An essential building block of our KDM secure encryption scheme is Yao's garbled circuit construction, attributed to [30]. Informally, the variant of this construction on which we rely transforms any circuit h with k input bits along with k pairs of random keys $(K_{1,0}, K_{1,1}), \ldots, (K_{k,0}, K_{k,1})$ into a "garbled circuit" GC such that the following properties hold:

- For any input $x \in \{0,1\}^k$ and any choice of $2k$ keys, the output $h(x)$ can be efficiently decoded (without knowing h) from GC and the k keys K_{i,x_i} corresponding to x.
- GC together with the k keys corresponding to x computationally hide all information about h other than the size of h and $h(x)$.
- GC alone computationally hides all information about h other than its size,

where the last two properties hold with respect to a random choice of the keys and a random execution of the transformation. The existence of a construction satisfying the above requirements is formally captured by the following theorem. See Appendix Cfor a derivation of this theorem from the literature.

Theorem 2.1 (Garbled circuits)**.** *Suppose that one-way functions exist. Then there is a pair of polynomial-time randomized algorithms* $(\mathsf{Garble}, \mathsf{GCEval})$ *that for security/input parameter k, output parameter m, and circuit size parameter s satisfy the following:*

Syntax. Garble *takes a $2k$ key tuple $\overline{K} = \{K_{i,b}\}_{i\in[k],b\in\{0,1\}}$, where $K_{i,b} \in \{0,1\}^k$, and a size s circuit describing a function $h : \{0,1\}^k \to \{0,1\}^m$, and outputs a "garbled circuit" GC.* GCEval *takes an input $x \in \{0,1\}^k$, a k key tuple, and a garbled circuit GC and outputs $y \in \{0,1\}^m$.*

Correctness. *We require that if* $GC = \mathsf{Garble}(\overline{K}, h)$ *then* $\mathsf{GCEval}(\overline{K}_x, GC) = h(x)$*, where we define* $\overline{K}_x = \{(x_i, K_{i,x_i})\}_{i\in[k]}$*.* [6]

Security against receiver. *For every polynomials* $s(k), m(k)$*, every* $x \in \{0,1\}^k$ *and every* $h, h' : \{0,1\}^k \to \{0,1\}^{m(k)}$ *of size* $s(k)$ *such that* $h(x) = h'(x)$*, if* $\overline{K}$ *is chosen at random then*

$$\overline{K}_x \circ \mathsf{Garble}(\overline{K}, h) \overset{c}{\approx} \overline{K}_x \circ \mathsf{Garble}(\overline{K}, h')$$

Security against outsiders. *For every polynomials* $s(k), m(k)$ *and every* $h, h' : \{0,1\}^k \to \{0,1\}^{m(k)}$ *of size* $s(k)$*, if* $\overline{K}$ *is chosen at random then*

$$\mathsf{Garble}(\overline{K}, h) \overset{c}{\approx} \mathsf{Garble}(\overline{K}, h')$$

2.2 Key-dependent message security

Loosely speaking, the notion of *key-dependent message (KDM) security* gives an adversary access to encryptions of messages of the form $h(sk)$, where $h : \mathcal{K} \to \mathcal{M}$ is a function that the adversary can choose from some family. The formal definition below is taken from Black et al. [6] and allows the function to depend on some $N = N(k)$ secret keys. While handling multiple keys is important for the application to formal cryptography (see Section 6), much of the technical challenge is already manifested in the case $N = 1$, and so the reader may wish to focus on this case initially.

Definition 2.2 (KDM security)**.** Let $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme with message space $\mathcal{M}$ and secret key space $\mathcal{K}$. Let $\overline{pk} := (\overline{pk}_1, \ldots, \overline{pk}_N)$ and $\overline{sk} := (\overline{sk}_1, \ldots, \overline{sk}_N)$ be public, resp., secret key vectors, where $N = N(k) > 0$ is a positive-valued function. Let A be a PPT machine. Let

- $\mathsf{Real}_{\overline{pk},\overline{sk}}$ be the oracle that on input a function $h : \mathcal{K}^N \to \mathcal{M}$ (encoded as a circuit) and $\mu \in [N]$ returns $C \leftarrow \mathsf{Enc}(\overline{pk}_\mu, h(\overline{sk}))$, and
- $\mathsf{Fake}_{\overline{pk}}$ be the oracle that on input h, μ as above returns $C \leftarrow \mathsf{Enc}(\overline{pk}_\mu, 0^k)$.

The *KDM advantage* of A is

$$\mathsf{Adv}^{\mathsf{KDM}}_{\mathsf{PKE},A}(k) := \mathsf{Pr}\left[A^{\mathsf{Real}_{\overline{pk},\overline{sk}}(\cdot,\cdot)}(\overline{pk}) = 1\right] - \mathsf{Pr}\left[A^{\mathsf{Fake}_{\overline{pk}}(\cdot,\cdot)}(\overline{pk}) = 1\right]$$

where $(\overline{pk}_i, \overline{sk}_i) \leftarrow \mathsf{Gen}(1^k)$ for $i \in [N]$ in both probabilities. We say that PKE is *KDM secure with respect to a function class* $\mathcal{H}$ iff for every polynomial N and every PPT A that only queries its oracle with functions $h \in \mathcal{H}$, the advantage function $\mathsf{Adv}^{\mathsf{KDM}}_{\mathsf{PKE},A}$ is negligible in the security parameter. PKE is *fully KDM secure* iff PKE is KDM secure with respect to the class $\mathcal{H}$ that consists of all functions.

Examples of KDM function classes. The following examples of KDM function classes will be important for us.

[6] For ease of notation we assume that the input x is included in the description of $\overline{K}_x$. This is needed to guarantee correctness even when a pair of keys happen to be identical. Alternatively, we could avoid giving x as input to GCEval by either settling for statistical correctness or allowing the keys to be correlated.

Clique/circular security. Let $\mathcal{S}_N$ consist of all functions $h_i : (\{0,1\}^k)^N \to \{0,1\}^k$ for $i \in [N]$, where $h_i(\overline{sk}_1, \ldots, \overline{sk}_N) = \overline{sk}_i$. Thus, KDM security with respect to $\mathcal{S}_N$ allows the adversary to obtain encryptions $\mathsf{Enc}_{\overline{pk}_i}(\overline{sk}_j)$ for every $i, j \in [N]$. This was called "clique security" by Boneh et al. [7] who gave a scheme that is KDM secure with respect to $\mathcal{S}_N$ for every N that is polynomial in the security parameter. (See Applebaum et al. [4] for another construction.) Security with respect to $\mathcal{S}_N$ implies "N-circular security". This notion, defined by [9] states that for independently generated N key pairs $(pk_1, sk_1), \ldots, (pk_N, sk_N)$, the vector of N encryptions $\mathsf{Enc}_{pk_1}(sk_2), \mathsf{Enc}_{pk_2}(sk_3), \ldots, \mathsf{Enc}_{pk_N}(sk_1)$ is indistinguishable from $\mathsf{Enc}_{pk_1}(0^k), \ldots, \mathsf{Enc}_{pk_N}(0^k)$.

Bounded security. Let $\mathcal{C}_{N,L}$ consist of all functions $h : (\{0,1\}^k)^N \to \{0,1\}^k$ that can be described with circuits of size at most L. We say that a scheme is (N, L) *bounded KDM secure*, if it is KDM secure with respect to $\mathcal{C}_{N(k),L(k)}$, where k denotes both the security parameter and the secret key size.[7]

Full (unbounded) security. Full KDM security is equivalent to requiring that a scheme is KDM secure with respect to $\mathcal{C}_{N,L}$ for every polynomials, in the security parameter, N and L. Note that this definition seems like the best one should look for, since a PPT adversary cannot generate circuits (i.e., queries) of superpolynomial size.

Finally, we say that a scheme has *single-key KDM security*, if in the KDM attack above the number of keys N is restricted to being 1. This notion makes sense with respect to bounded/unbounded security, where in the case of or clique or circular security it is equivalent to "self reference security" — the adversary has access to $\mathsf{Enc}_{pk}(sk)$.

2.3 KDM security from homomorphic encryption

In this section we observe that one can get KDM security from a certain kind of homomorphic encryption schemes.

Definition 2.3 (Homomorphic encryption)**.** Let $\mathcal{H} = \{\mathcal{H}_k\}$ be a sequence of sets of Boolean circuits. A tuple of algorithms $\xi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ is a *homomorphic encryption scheme* with respect to $\mathcal{H}$, if $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is a public key encryption scheme, and in addition for every $(pk, sk) \leftarrow \mathsf{Gen}(1^k)$, $h \in \mathcal{H}_k$ and message $M \in \mathcal{M}$

$$\mathsf{Dec}_{sk}\big(\mathsf{Eval}_{pk}(h, \mathsf{Enc}_{pk}(M))\big) = h(M)$$

We say that ξ satisfies *strong (statistical) function-privacy* if for every $h \in \mathcal{H}_k$, pk in the range of $\mathsf{Gen}(1^k)$ and $M \in \mathcal{M}$, $\mathsf{Eval}_{pk}(h, \mathsf{Enc}_{pk}(M)) \stackrel{s}{\approx} \mathsf{Enc}_{pk}(h(M))$.

We say that ξ satisfies *weak (statistical) function-privacy* if for every $h, h' \in \mathcal{H}_k$, pk in the range of $\mathsf{Gen}(1^k)$ and $M \in \mathcal{M}$, if $h(M) = h'(M)$ then $\mathsf{Eval}_{pk}(h, \mathsf{Enc}_{pk}(M)) \stackrel{s}{\approx} \mathsf{Eval}_{pk}(h', \mathsf{Enc}_{pk}(M))$.[8]

We say that a scheme is *fully homomorphic* if **(1)** for every polynomial $s = s(k)$ it is homomorphic with respect to the family $\mathcal{H} = \{\mathcal{H}_k\}$, where $\mathcal{H}_k$ is the set of all Boolean circuits of size at most

[7] Requiring the secret key to be at most k prevents trivialities such as making the key so big that $L(k)$-sized circuits don't have time to read it. In fact, our scheme will satisfy a slightly stronger notion which is that the key generation algorithm will be completely independent of N and L, see Definition 6.1.

[8] One can naturally define *computational* versions of weak and strong function privacy, in which case one needs to allow the distinguisher to get the secret key as part of the input, and in some applications also the randomness used to generate this secret key.

$s(k)$, and **(2)** the running time (and hence also output size) of both the encryption and decryption algorithm is a fixed polynomial in the security parameter k. It was a longstanding open problem to come up with even a plausible candidate for such a scheme, until this was achieved this year by Gentry [16], who gave such a candidate based on ideal lattices.[9] If a scheme satisfies only **(1)** (but not necessarily **(2)**) then we say that it is *size-dependent* homomorphic encryption. There is a trivial construction of a size dependent homomorphic encryption: just have Eval concatenate the circuit to the ciphertext. Using Yao's garbled circuit construction and two-message OT one can get a size-dependent homomorphic encryption with weak function privacy. In contrast, *strong* function privacy for this class $\mathcal{H}$ implies condition **(2)**.

As mentioned in Section 1.2, we observe that a homomorphic encryption scheme with respect to a class $\mathcal{H}$ that is strongly function-private and is circular secure, is also KDM secure with respect to the same class. This already implies that Gentry's scheme is fully KDM secure under certain assumptions that do not refer to *full* KDM security (i.e., hardness of a certain bounded-distance decoding problem on ideal lattices, a sparse subset sum problem, and assuming the scheme is *circular* secure). Moreover, for this application we can relax the condition to *weak* function-privacy:

Theorem 2.4. *Suppose that there is a homomorphic encryption scheme with respect to a class $\mathcal{H}$ that is weakly function private and 1-circular secure. Then there is a single-key KDM-secure scheme with respect to the same class $\mathcal{H}$.*

Proof Sketch Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be the homomorphic encryption scheme. Our encryption scheme $(\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ will be as follows:

Key Generation $\mathsf{Gen}'(1^k)$ runs $(pk, sk) \leftarrow \mathsf{Gen}(1^k)$ and outputs the same secret key sk and as public key the concatenation of pk and $C = \mathsf{Enc}_{pk}(sk)$.

Encryption $\mathsf{Enc}'_{pk,C}(M)$ outputs $\mathsf{Eval}_{pk}(const^M, C)$, where $const^M$ is the constant function that always outputs M.

Decryption We have $\mathsf{Dec}' = \mathsf{Dec}$.

Correctness follows easily from the homomorphic property. For security, consider a KDM attacker, that queries an oracle with h and gets back $\mathsf{Enc}'_{pk,C}(h(sk)) = \mathsf{Eval}_{pk}(const^{h(sk)}, C)$. We proceed by a hybrid argument. Suppose that the oracle was changed so that it returned $\mathsf{Eval}_{pk}(h, C)$. Since C is an encryption of sk, and obviously $const^{h(sk)}(sk) = h(sk)$, and the scheme satisfies weak function-privacy this will not change the attacker's output distribution. (Since we need the secret key to compute $const^{h(sk)}$, we will need here to use the fact that weak function-privacy holds even with respect to distinguishers that know the secret key.) The new oracle, however, can be simulated by the attacker on its own (since it does not use the secret key at all, but only h and C). Hence, we complete the proof by appealing to the circular security of the encryption, to argue that C might as well be an encryption of "junk". ∎

As a corollary, assuming the *circular* security of a version of Paillier's cryptosystem [28, 12], the homomorphic PKE construction from [21] yields a KDM-secure encryption scheme with respect to

[9] The fully homomorphic version of Gentry's scheme requires three assumptions: hardness of a certain bounded-distance decoding problem on ideal lattices, hardness of a sparse version of subset sum, and circular security of his basic ideal-lattice based scheme.

the class of branching programs of a bounded (polynomial) *length*, but unbounded (polynomial) *size*. In other words, the length of the ciphertexts should only depend on the length of branching programs computing the KDM function but not on their size. Compared to the alternative based on the circular-secure version of Gentry's scheme, the conclusion is much weaker but the assumption is different (and seemingly more conservative).

3 Targeted Encryption

The main tool we use to realize our KDM secure scheme is a new notion we call *targeted encryption*. This is a variant of a public key encryption scheme that has the following curious property: the encryption algorithm gets, apart from the message x, two additional inputs: an index $i \in [k]$ (where k is the bit length of the secret key), and a bit b. The decryption algorithm successfully retrieves x if the i^{th} bit of the secret key is b, but otherwise gets no information about x.[10]

Definition 3.1 (Targeted encryption)**.** An *targeted encryption scheme* TES consists of a tuple of algorithms $(\mathsf{TGen}, \mathsf{TEnc}, \mathsf{TDec})$ such that on security parameter k, TGen outputs a pair (pk, sk) with $sk = (sk_1, \ldots, sk_k) \in \{0,1\}^k$ and:

Targeted decryption For every message $x \in \{0,1\}^n$ and index $i \in [k]$,

$$\mathsf{TDec}_{sk}(\mathsf{TEnc}_{pk,i,sk_i}(x)) = x \ .$$

I.e., it is possible for a sender, given (i, b), to encrypt a message x such that the following hold: if the i^{th} bit of the secret key is b, then the receiver decrypts this message successfully.

(Statistical) security against receiver For every $x, x' \in \{0,1\}^n$ and index $i \in [k]$,

$$\mathsf{TEnc}_{pk,i,1-sk_i}(x) \overset{\text{s}}{\approx} \mathsf{TEnc}_{pk,i,1-sk_i}(x') \ .$$

I.e., if the i^{th} bit of the secret key is not b, then the receiver gets no information about the message x.[11]

Security against outsiders For every $x, x' \in \{0,1\}^n$, index $i \in [k]$, and $b \in \{0,1\}$,

$$pk \circ \mathsf{Enc}_{pk,i,b}(x) \overset{\text{c}}{\approx} pk \circ \mathsf{Enc}_{pk,i,b}(x') \ .$$

I.e., outsiders, who do not know the secret key, get no information about the encryption, even if the i^{th} bit of sk does equal b.

The next theorem states that targeted encryption scheme can be obtained from either the DDH or the LWE assumptions.

[10] We do not actually require a targeted encryption to also have a standard ("untargeted") encryption algorithm, that always succeeds although this can easily be achieved by, say, concatenating two encryptions using parameters $i, 0$ and $i, 1$. Later, to achieve multiple-key security, we will need to assume such an algorithm with particular properties, see Section 5.

[11] For our purposes we can relax this notion to computational indistinguishability with respect to distinguishers that get the secret key as additional input.

Theorem 3.2. *Suppose that (1) the DDH Assumption holds,* or *(2) the LWE assumption holds (with certain parameters),*[12] *then there exists a targeted encryption scheme.*

Theorem 3.2 is proven by showing that targeted encryption is implied by both the work of Boneh et al. [7] and the work of Applebaum et al. [4] (see appendices A and Bfor the formal proof). The idea of the proof is as follows. Both works give schemes that are KDM secure with respect to affine functions over $\mathbb{Z}_q^k$ for some number q, where k being the secret key size. Their proofs,[13] however, actually give the following stronger homomorphic property: there exists an algorithm Eval that gets the public key and an affine function $h : \mathbb{Z}_q^k \to \mathbb{Z}_q^k$, and outputs an encryption of $h(sk)$ that is statistically indistinguishable from $\mathsf{Enc}_{pk}(h(sk))$. Note that this is a property that indeed immediately implies KDM security for affine functions. We will use this property to get the following targeted encryption scheme: to encrypt a message $x \in \{0,1\}^n$ so that it can only be decrypted if the i^{th} bit of the key is b, we view x as an element inside $\mathbb{Z}_q$ (using some natural embedding for large enough q, where if n is too large, we encrypt x in chunks) and choose a random $r \in \mathbb{Z}_q$ and use the encryption of scheme to encrypt $r \cdot (sk_i - b) + x$. Note that this is an affine function of sk,[14] and its value is independent of x if $sk_i \neq b$, but is equal to x otherwise. Some complications arise from the fact that in [7] the group is actually given "in the exponent", where in [4] the key is not a bit string, but rather a vector in $\mathbb{Z}_q^k$. Nevertheless, these issues can be easily handled in both cases.

Discussion — Targeted encryption and oblivious transfer. Recall that in a (one out of two) *oblivious transfer* (OT) protocol, a sender holds a pair of values (x_0, x_1), and a receiver has a bit b. At the end of the protocol, the receiver learns x_b and learns nothing about x_{1-b}, while the sender learns nothing about b. A *two-message* OT protocol is one that consists of only two messages — the first from the receiver and the second from the sender. It is easy to see that any two-message OT implies a public-key cryptosystem (with the first message being the public key); in addition, almost all popular candidates for public-key cryptosystems imply two-message OT protocols.

A targeted encryption scheme can be thought of as a type of "self-referential" OT where the receiver's input selection bits are equal to the secret information it keeps after its first message (i.e., the secret key). It does not satisfy, however, the standard notion of OT, since the sender is not guaranteed to learn *nothing* about this secret key (although the "security against outsiders" property does imply that the sender cannot recover it completely). We note that it is possible (though we do not need to use this fact in this paper) to transform an OT with such a guarantee into a full-fledged OT, using the techniques of [17, 11].

4 Our Bounded KDM Secure Construction

Let k be the security parameter. Let $\mathsf{TES} = (\mathsf{TGen}, \mathsf{TEnc}, \mathsf{TDec})$ be a targeted encryption scheme. We will construct the following PKE scheme $\mathsf{bKDM} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ that is parameterized over polynomials N and L.

Key generation. $\mathsf{Gen}(1^k)$ samples and outputs $(pk, sk) \leftarrow \mathsf{TGen}(1^k)$.

[12] The exact group for DDH and parameters for LWE are inherited from the assumptions [7, 4]; one important note is that we need to assume LWE for a prime modulus that is polynomial in the security parameter. See appendices A and B for more details.

[13] In [7]'s case, the above is true for what they call their "expanded" scheme.

[14] Indeed this is the function $h(sk) = \langle \overline{r}, sk \rangle + x'$, where $\overline{r}_i = r$, $\overline{r}_j = 0$ for $j \neq i$, and $x' = x - b \cdot r$.

Encryption. $\mathsf{Enc}_{pk}(M)$ chooses $2k$ random strings $\overline{K} = (K_{i,b})_{(i,b)\in[k]\times\{0,1\}}$ and computes the garbled circuit transformation on $\overline{K}$ and the constant function $const^M$ that outputs M on *every* input $x \in \{0,1\}^k$. We use S, which is some fixed polynomial $S(N,L)$ to be specified later, as the size parameter for the garbled circuit transformation. Let GC be the resulting output. Enc also computes for every $(i,b) \in [k] \times \{0,1\}$ the value $\tilde{K}_{i,b} = \mathsf{TEnc}_{pk,i,b}(K_{i,b})$ and outputs $C := (GC, (\tilde{K})_{i,b})_{(i,b)\in[k]\times\{0,1\}})$ as the ciphertext.

Decryption. $\mathsf{Dec}_{sk}(GC, (\tilde{K}_{i,b})_{i,b})$ parses $sk = (sk_1, \ldots, sk_k) \in \{0,1\}^k$ and computes the value $K_i = \mathsf{TDec}_{sk}(\tilde{K}_{i,sk_i})$ for every $i \in [k]$. Then, it outputs the result of evaluating the garbled circuit GC on $K_1, \ldots, K_k$.

It is easily verified that the decryption will indeed output $const^M(sk) = M$. We would like to emphasize that key generation does *not* depend on L or N, only encryption does. Hence, we can generate and distribute keys even without knowing L and N in advance.

4.1 Single-key security of the construction

We now show that bKDM is KDM secure for the case of a single key (i.e., $N = 1$). In Section 5, we show that if the underlying targeted encryption scheme is circular secure (when suitably interpreted as a PKE scheme), bKDM actually is secure for an arbitrary number of keys.

Theorem 4.1. *If* $\mathsf{TES} = (\mathsf{TGen}, \mathsf{TEnc}, \mathsf{TDec})$ *is a targeted encryption scheme, then for every polynomial* L, bKDM *instantiated with* $S(N,L) = L$ *is* $(1,L)$*-bounded KDM secure.*

Proof Fix $N = 1$, an arbitrary L and a PPT adversary A on bKDM's bounded KDM security. In order to keep the notations simple, we concentrate on the single query case (i.e., the attacker only asks a single key related query). The multi query case, however, easily follows from the same lines. We proceed in games. Let X_i be A's output in Game i, and write $X_i \approx X_j$ as a shorthand for $\mathsf{Pr}[X_i = 1] - \mathsf{Pr}[X_j = 1] \in \mathsf{negl}$. See Table 1 for an overview of all games used in the proof. In all the following games, (sk, pk) are chosen at random using TGen and the oracles get $h : \{0,1\}^k \to \{0,1\}^k$ as input.

Game 0 is the real KDM game. Namely, the oracle $\mathsf{Real}_{pk,sk}$ returns the ciphertext $\mathsf{Enc}_{pk}(h(sk))$. Recall that this is computed by (1) choosing a random $2k$ key tuple $\overline{K}$, (2) encrypting the keys using TEnc to obtain a tuple of ciphertexts $\tilde{K}$ where $\tilde{K}_{i,b} = \mathsf{TEnc}_{pk}(K_{i,b})$ for every $(i,b) \in [k] \times \{0,1\}$, and (3) computing $GC = \mathsf{Garble}(\overline{K}, const^{h(sk)})$. Ciphertext is $C := (GC, \tilde{K})$.

In **Game** 1 the oracle sets $\tilde{K}_{i,b}$ to $\mathsf{TEnc}_{pk}(0^k)$, instead of $\mathsf{TEnc}_{pk}(K_{i,b})$, for every (i,b) with $sk_i \neq b$. (Note that we still use the original $\overline{K}$ in the garbled circuit construction.) Since GC is independent from the random coins used to encrypt $\tilde{K}$, the "security against receiver" property of TES yields that $X_0 \approx X_1$.

In **Game** 2 the oracle uses h instead of $const^{h(sk)}$ in the garbled circuit construction (i.e., it computes $GC = \mathsf{Garble}(\overline{K}, h)$). Since $h(sk) = const^{h(sk)}(sk)$ and only the keys $\overline{K}_{sk} = (\overline{K}_{i,sk_i})_i$ are used outside the garbled circuit construction, the security against receiver of the garbled circuit construction yields that $X_1 \approx X_2$. We note that the only role of the secret key in this game, is for deciding which elements of $\tilde{K}$ are replaced by encryptions of 0^k.

In **Game** 3 we go back to using the original $\tilde{K}$ (also for the (i,b) with $b \neq sk_i$). Again, the "security against receiver" property of TES implies that $X_2 \approx X_3$. Note that in this game the encryption oracle does not use the secret key *at all*.

We define **Game** 4 to be the variant of **Game** 3 in which we set $\tilde{K}_{i,b} = \mathsf{TEnc}_{pk}(0^k)$ for every i, b (i.e., we ignore the value of $\overline{K}$ for this part). Since the secret key is never used in either Game 3 or Game 4, the "security against outsiders" property of the TES implies that $X_3 \approx X_4$. Note that in this game, the vector $\overline{K}$ is independent of $\tilde{K}$.

In **Game** 5 we change h to $const^{0^k}$ in the garbled circuit construction. Since no information on the key vector $\overline{K}$, except for the garbled circuit itself, is given in both oracles, the "security against outsiders" property of the garbled circuit construction implies that $X_4 \approx X_5$. (Note that we need to use the "security against outsiders" and not the "security against receiver" property, since obviously we have no guarantee that $h(sk) = const^{0^k}(sk)$.)

We define **Game** 6 to be the game in which we go back to using the real $\tilde{K}$. Since the oracles do not use the secret key, we get that $X_5 \approx X_6$. Observe that the encryption oracle is exactly the Fake oracle as per Definition 2.2, and hence we have completed the proof. ∎

Game	Oracle needs	$\tilde{K}_{i,sk_i}$	$\tilde{K}_{i,1-sk_i}$	Function in GC	Remark
0	pk, sk	$\mathsf{TEnc}(K_{i,sk_i})$	$\mathsf{TEnc}(K_{i,1-sk_i})$	$const^{h(sk)}$	Real KDM game
1	pk, sk	$\mathsf{TEnc}(K_{i,sk_i})$	$\boxed{\mathsf{TEnc}(0^k)}$	$const^{h(sk)}$	Security against receiver of TES
2	pk, sk	$\mathsf{TEnc}(K_{i,sk_i})$	$\mathsf{TEnc}(0^k)$	$\boxed{h}$	Security against receiver of GC
3	$\boxed{pk}$	$\mathsf{TEnc}(K_{i,sk_i})$	$\boxed{\mathsf{TEnc}(K_{i,1-sk_i})}$	h	Security against receiver of TES
4	pk	$\boxed{\mathsf{TEnc}(0^k)}$	$\boxed{\mathsf{TEnc}(0^k)}$	h	Security against outsiders of TES
5	pk	$\mathsf{TEnc}(0^k)$	$\mathsf{TEnc}(0^k)$	$\boxed{const^{0^k}}$	Security against outsiders of GC
6	pk	$\boxed{\mathsf{TEnc}(K_{i,sk_i})}$	$\boxed{\mathsf{TEnc}(K_{i,1-sk_i})}$	$const^{0^k}$	Security against outsiders of TES
					fake KDM game

Table 1: Overview of the games used in the proof of Theorem 4.1. We use $\boxed{\text{boxes}}$ to highlight the component that changed from the previous hybrid, and note in the remark the justification for the fact that the hybrid is indistinguishable from the previous one.

5 Multiple Key Security

While the notion of KDM security for a single key is challenging and elegant, many of the applications actually require KDM security in the presence of arbitrarily (polynomially) many keys. Hence, let now the number of keys N be an arbitrary polynomial in the security parameter. We will prove that our scheme bKDM from Section 4 is (N, L)-bounded KDM secure, but now under an additional assumption, and with different parameters.

Complication and central idea. Recall the proof of Theorem 4.1. There, we have first substituted the function $const^{h(sk)}$ that is evaluated by GC by the KDM query function h itself. By the secrecy against receiver property of the garbled circuit, we could argue that this change goes unnoticed by the receiver. This modification was a crucial step in our proof, since it allowed to construct the garbled circuit without knowing sk. Recall that in multiple secret keys case, the security is defined with respect to N public and secret keys pairs $((\overline{pk}_1, \overline{sk}_1), \ldots, (\overline{pk}_N, \overline{sk}_N))$, and the adversary gets encryptions of a query function $h = h(\overline{sk})$ under arbitrary $\overline{pk}_\mu$ for $\mu \in [N]$. Hence, we cannot simply substitute $const^{h(\overline{sk})}$ with h directly (the secrecy against receiver property

of the garbled circuit would not help in this case, since we cannot claim that $h(\overline{sk}_\mu) = const^{h(\overline{sk})}$).[15] Instead, we will substitute $const^{h(\overline{sk})}$ with a function h' for which $h'(\overline{sk}_\mu) = h(\overline{sk})$. This function h' contains an encryption of $\overline{sk}$ under the receiver's public key $\overline{pk}_\mu$. In this, we will have to interpret bKDM's underlying targeted encryption scheme TES as a *circular-secure* encryption scheme. (Circular security is required to guarantee that we can later replace these encryptions of secret keys with 0^k-encryptions.) Since our targeted encryption scheme instance is based on the clique-secure encryption schemes of [7, 4], it already has this property. The remaining part of the proof follows the proof of Theorem 4.1.

Definition 5.1 (Augmented targeted encryption)**.** An *augmented targeted encryption scheme* $\mathsf{ATES} = (\mathsf{TGen}, \mathsf{TEnc}, \mathsf{TDec}, \mathsf{Enc}, \mathsf{Dec})$ is a targeted encryption scheme $(\mathsf{TGen}, \mathsf{TEnc}, \mathsf{TDec})$, complemented by PPT algorithms Enc, Dec for (un-targeted) encryption and decryption. We require that $(\mathsf{TGen}, \mathsf{Enc}, \mathsf{Dec})$ is a public-key encryption scheme with message space $\mathcal{M} \subseteq \{0,1\}^k$. In particular, $\mathsf{Dec}_{sk}(\mathsf{Enc}_{pk}(M)) = M$ for all $(pk, sk) \leftarrow \mathsf{TGen}(1^k)$ and $M \in \{0,1\}^k$.

We say that ATES is circular secure if $(\mathsf{TGen}, \mathsf{Enc}, \mathsf{Dec})$ is. We stress that our both TES instances from Appendices A and B are circular secure augmented targeted encryption schemes with the natural encryption and decryption algorithms from Boneh et al. [7] and Applebaum et al. [4] respectively. The following theorem implies our main result (i.e., Theorem 1.1).

Theorem 5.2. *If* $\mathsf{ATES} = (\mathsf{TGen}, \mathsf{TEnc}, \mathsf{TDec}, \mathsf{Enc}, \mathsf{Dec})$ *is a circular secure augmented targeted encryption scheme, then for every polynomials* L *and* N*,* bKDM *instantiated with a suitable polynomial* $S(N, L)$ *is bounded KDM secure.*

Proof Fix arbitrary N and L, and a PPT adversary A on bKDM's bounded KDM security. As in Section 4, we consider the single query case. Again, we proceed in games. Our game are analogous to the ones used in the proof of Theorem 4.1, except that we add one new game, Game 3.5, that uses the circular-security property of the augmented targeted encryption scheme. As in that proof, we let X_i be A's output in Game i, and write $X_i \approx X_j$ as a shorthand for $\mathsf{Pr}[X_i = 1] - \mathsf{Pr}[X_j = 1] \in \mathsf{negl}$. See Table 2 for an overview of all games used in the proof. We will be very brief in the parts that are identical to the proof of Theorem 4.1. In all the following games, $((\overline{pk}_1, \overline{sk}_1), \ldots, (\overline{pk}_N, \overline{sk}_N))$ are chosen independently at random using Gen, and the oracles get $(h : (\{0,1\}^k)^N \to \{0,1\}^k, \mu \in [N])$ as input.

Game 0 is the real KDM game. Namely, the oracle $\mathsf{Real}_{\overline{pk},\overline{sk}}$ returns the ciphertext $\mathsf{Enc}_{\overline{pk}_\mu}(h(\overline{sk})) = (GC, \tilde{K})$ for encrypted keys $\tilde{K}$ and a garbled circuit GC that evaluates $const^{h(\overline{sk})}$.

In **Game** 1 the oracle sets $\tilde{K}_{i,b}$ to encryption of 0^k instead of encryption of the key $K_{i,b}$, for every (i, b) with $\overline{sk}_{\mu,i} \neq b$, where $\overline{sk}_\mu = (\overline{sk}_{\mu,1}, \ldots, \overline{sk}_{\mu,k})$. The "security against receiver" property of ATES yields that $X_0 \approx X_1$.

In **Game** 2 the oracle first computes ATES ("untargeted") encryptions $C_i \leftarrow \mathsf{Enc}_{\overline{pk}_i}(\overline{sk}_{i \bmod N+1})$ for every $i \in [N]$. It then constructs a circuit $h'_{\overline{C}}$ with hardwired $\overline{C} := (C_i)_{i \in [N]}$ and h that, on input $\overline{sk}_\mu$, proceeds as follows: it (1) for every $i \in [N]$ retrieves $\overline{sk}_{(i \bmod N)+1} \leftarrow \mathsf{Dec}_{\overline{sk}_i}(C_i)$, starting from $i = \mu$ till $\mu - 1$ (till N if $\mu = 1$), and (2) computes and outputs $h(\overline{sk})$. The KDM encryption oracle then acts as the one of Game 1, but replaces the function $const^{h(\overline{sk})}$ that GC evaluates with $h'_{\overline{C}}$. We note that $h'_{\overline{C}}(\overline{sk}_\mu) = const^{h(\overline{sk})}(\overline{sk}_\mu) = h(\overline{sk}_\mu)$, and that only the keys $\overline{K}_{\overline{sk}_\mu}$ are ever used

[15] $h(\overline{sk}_\mu)$ is not even well defined; h is expecting a vector of (secret) keys as input, and not a single key.

outside the garbled circuit construction. We also note that $h'_{\overline{C}}$ can be encoded as a circuit of size $S(N, L)$, for some polynomial S. (This S will be the size function with which we instantiate bKDM, see the theorem's statement). Hence, $X_1 \approx X_2$ by the security against receiver of the garbled circuit construction.

In **Game** 3 we go back to the real $\tilde{K}$. Again, the "security against receiver" property of ATES implies that $X_2 \approx X_3$. Note that at this point, the oracle uses the secret keys $\overline{sk}$ only to construct the encryptions $(C_i \leftarrow \mathsf{Enc}_{\overline{pk}_\mu}(\overline{sk}_i))_{i\in[N]}$.

In **Game** 3.5 we replace the ciphertext vector $\overline{C}$ that is hardwired into $h'_{\overline{C}}$ with a vector $\overline{C}_0 = (C_i^0)_{i\in[N]}$ of 0^k-encryptions (i.e., $C_i^0 \leftarrow \mathsf{Enc}_{\overline{pk}_\mu}(0^k)$). Since $\overline{sk}$ was only used to construct $\overline{C}$, the circular security of ATES implies that $X_3 \approx X_{3.5}$. Note that the oracle in Game 3.5 only uses $\overline{pk}$, but not $\overline{sk}$. It also worth mentioning, that this is the only place in the proof where the assumed circular security of ATES is used.

In **Game** 4 we set $\tilde{K}_{i,b} = \mathsf{TEnc}_{\overline{pk}_\mu}(0^k)$ for *every* (i, b) (i.e., we ignore the value of $\overline{K}$ for this part). Since neither Game 3.5 nor Game 4 use the secret keys $\overline{sk}$, the "security against outsiders" property implies $X_3 \approx X_4$.

In **Game** 5 we change $h'_{\overline{C}_0}$ to $const^{0^k}$ in the garbled circuit construction. The "security against outsiders" property of the garbled circuit construction implies that $X_4 \approx X_5$, since the key vector $\overline{K}$ is only used to construct GC in both games.

In **Game** 6 we are back again to original $\tilde{K}$. Since the oracles do not use $\overline{sk}$, we get $X_5 \approx X_6$. Observe that this encryption oracle is exactly the Fake oracle as per Definition 2.2, nd hence we have completed the proof. ∎

Game	Oracle needs	$\tilde{K}_{i,\overline{sk}_{\mu,i}}$	$\tilde{K}_{i,1-\overline{sk}_{\mu,i}}$	Function in GC	Remark
0	$\overline{pk}, \overline{sk}$	$\mathsf{TEnc}(K_{i,\overline{sk}_{\mu,i}})$	$\mathsf{TEnc}(K_{i,1-\overline{sk}_{\mu,i}})$	$const^{h(\overline{sk})}$	Real KDM game
1	$\overline{pk}, \overline{sk}$	$\mathsf{TEnc}(K_{i,\overline{sk}_{\mu,i}})$	$\boxed{\mathsf{TEnc}(0^k)}$	$const^{h(\overline{sk})}$	Security against receiver of ATES
2	$\overline{pk}, \overline{sk}$	$\mathsf{TEnc}(K_{i,\overline{sk}_{\mu,i}})$	$\mathsf{TEnc}(0^k)$	$\boxed{h'_{\overline{C}}}$	Security against receiver of GC
3	$\overline{pk}, \overline{sk}$	$\mathsf{TEnc}(K_{i,\overline{sk}_{\mu,i}})$	$\boxed{\mathsf{TEnc}(K_{i,1-\overline{sk}_{\mu,i}})}$	$h'_{\overline{C}}$	Security against receiver of ATES
3.5	$\boxed{\overline{pk}}$	$\mathsf{TEnc}(K_{i,\overline{sk}_{\mu,i}})$	$\mathsf{TEnc}(K_{i,1-\overline{sk}_{\mu,i}})$	$\boxed{h'_{\overline{C}_0}}$	Clique security of ATES
4	$\overline{pk}$	$\boxed{\mathsf{TEnc}(0^k)}$	$\boxed{\mathsf{TEnc}(0^k)}$	$h'_{\overline{C}_0}$	Security against outsiders of ATES
5	$\overline{pk}$	$\mathsf{TEnc}(0^k)$	$\mathsf{TEnc}(0^k)$	$\boxed{const^{0^k}}$	Security against outsiders of GC
6	$\overline{pk}$	$\boxed{\mathsf{TEnc}(K_{i,\overline{sk}_{\mu,i}})}$	$\boxed{\mathsf{TEnc}(K_{i,1-\overline{sk}_{\mu,i}})}$	$const^{0^k}$	Security against outsiders of ATES
					fake KDM game

Table 2: Overview of the games used in the proof of Theorem 5.2.

6 Application to formal cryptography

One of the main motivations to study KDM security lies in the connection between formal and computational cryptography. In formal cryptography (starting with [13, 14, 25]), cryptographic operations like encryption or digital signatures are abstracted as symbolic operators that (only)

adhere to natural rules like $D_K(E_K(M)) = M$ for symmetric encryption and decryption operators E and D. A simple calculus like this enables machine-assisted security analysis (e.g., [22, 24]). It is not a priori clear, however, that security properties proved in the symbolic calculus also hold for the computational implementation of the protocol.

Computational soundness. Abadi and Rogaway [1] were the first to relate the formal and computational views on cryptography. Specifically, they showed that every symbolically proven property also holds in the computational world, assuming a suitable computational implementation. This is usually referred to as a soundness result, and suitable computational implementations are dubbed sound. In order to provide computational soundness in this sense in face of a passive adversary, an encryption scheme essentially needs to be IND-CPA secure.

Key-cyclic expressions. There is a technical nuisance, however, that limits the generality and expressivity of [1]'s approach. Namely, the soundness result only holds for protocols that do *not* contain key-cyclic expressions. That is, only protocols in which no expressions with cyclic dependencies of encryption keys (such as $E_{K_1}(E_{K_2}(K_1))$) appear are considered. This is for the following reason: in the symbolic setting, the natural deduction rules explicitly require secret keys for decryption. Hence, the encrypted plaintexts in such expressions are secret by definition in the symbolic world (i.e., there is no formal way to apply, say, D_{K_2} on the ciphertext $E_{K_1}(E_{K_2}(K_1))$). On the other hand, key-dependent messages like the one above, are not modeled in standard (computational) security experiments for encryption schemes.[16] Hence, there is an asymmetry between symbolic and computational setting, and any soundness result that connects symbolic encryption and *standard* computational encryption notions has to exclude key-cyclic expressions.

Soundness from Bounded KDM Security. It was informally claimed by Black et al. [6], and formally proven by Adão et al. [2], that *fully* KDM-secure encryption schemes imply computational soundness for *arbitrary* symbolic protocols. Since we can only achieve bounded KDM security against arbitrary circuits up to a certain size, we ask whether bounded KDM security suffices for computational soundness of arbitrary symbolic protocols. The answer we give is essentially affirmative.

To do so, we introduce the following slight strengthening of bounded KDM security:

Definition 6.1 (Length-dependent bounded KDM security)**.** A PKE scheme with message space $\mathcal{M} \subseteq \{0,1\}^*$ is *N-key length-dependent bounded KDM secure*, if it is KDM secure with respect to the circuit class of all $h : (\{0,1\}^k)^N \to \{0,1\}^{\lfloor\sqrt{|h|}\rfloor}$, where $|h|$ is the circuit size of h.

That is, length-dependent KDM secure schemes are secure against larger KDM queries if longer messages are encrypted. We stress that our scheme bKDM from Section 4 is N-key length-dependent bounded KDM secure, if we choose L suitably (e.g., $L = |M|^{2.1}$) during encryption. Namely, bKDM's key generation algorithm does not depend on N or L, and the proofs of Theorems 4.1 and 5.2 do not use that L is fixed.

Theorem 6.2 (Following [1, 2]: Bounded KDM security implies soundness)**.** *Let* bKDM *be an N-key length-dependent KDM secure PKE scheme, and let P be a symbolic protocol with N parties in the setting of Adão et al. [2]. Then* bKDM *provides a computationally sound implementation of P.*

[16] Some subsequent soundness results (e.g., [5, 26]) consider an active adversary and require IND-CCA security. We stress that does not change the technical complications regarding key-cyclic expressions.

We stress that the choice of symbolic setting [2] was made only for simplicity.

Proof outline Without loss of generality, we assume that bKDM-encryptions are padded to have bit length at least the square root of the number of steps that encryption takes.

For soundness in the sense of [2], we need to prove that for all formal expressions M that occur in protocol P, we have $[\![M]\!] \overset{c}{\approx} [\![\mathsf{pattern}(M)]\!]$. Here, $[\![M]\!]$ denotes the computational implementation of M, i.e., the result of translating M into a bit string using the given encryption scheme. Let $\mathsf{pattern}(M)$ denote the formal expression one gets when replacing all subexpressions of M that are "inaccessible" (i.e., cannot be derived using the deduction relation and publicly available information/keys) by a special symbol $\Box$. Encryptions of $\Box$ are translated into 0-encryptions of appropriate length.

The proof from [2], building on [1], shows that $[\![M]\!] \overset{c}{\approx} [\![\mathsf{pattern}(M)]\!]$ using a reduction to (full) KDM security. Concretely, they employ an algorithm CONVERT that uses bKDM's encryption algorithm and a (real or fake) KDM encryption oracle. CONVERT recursively translates M into a bit string such that when CONVERT uses the real KDM oracle, $[\![M]\!]$ is the result, and when CONVERT uses the fake KDM oracle, $[\![\mathsf{pattern}(M)]\!]$ is the result. CONVERT uses its KDM oracle only for encryptions with respect to "invisible" keys that cannot be symbolically deduced from M. For proving Theorem 6.2, we only need to check that all such KDM queries can be expressed as functions in $\mathcal{C}_{N,L}$, for suitable polynomial N, L.

Without loss of generality, we assume that M is an encryption of some term M' under some public key K with invisible (in the above sense) secret key K^{-1}. We may also assume that M' contains no decryption operators. Namely, decryption either yields the original plaintext (in case decryption gets as input an encryption with respect to a matching key, so encryption and decryption cancel out), or an invalid symbol (in case of non-matching keys or a non-ciphertext input, which can be decided from the structure of M'). We can now argue inductively that the square of the bit length of M' is an upper bound for the time it takes to compute M' recursively from all its subterms. For encryptions inside M', this holds by our assumption on bKDM, and for subterms of other type, we assume a suitable computational implementation.[17] Hence, we can (possibly recursively) build a KDM function h of size L which is at most the square of the bit length of M'. The length-dependent KDM security of bKDM yields that $[\![M]\!] \overset{c}{\approx} [\![\mathsf{pattern}(M)]\!]$, and soundness follows. ∎

Application of our results. Theorem 6.2 can be instantiated with our scheme bKDM from Section 4. (As argued above, bKDM actually is N-key length-dependent bounded KDM secure.) This yields the first encryption scheme that provides soundness under a standard computational assumption.

Relation to circular security and extensibility. For extremely simple calculi that only feature public-key encryption, along with a few syntactic operations like pairings of terms, already clique security may enable soundness. (This is so since all key-dependent encryptions that can possibly occur in a symbolic protocol can be traced back to simple encryptions of the form $E_{K_i}(K_j)$, assuming a suitable way to encrypt longer terms in chunks.) Nevertheless, we stress that our scheme bKDM allows much richer classes of calculi. For instance, the above soundness proof also works in the presence of signatures, so that terms of the form $E_{K_i}(sig(K_j), K_\ell)$ may occur. (The crucial

[17] For instance, although the setting of [2] only contains encryption, we would assume that signatures are suitably padded. As with decryption, we can argue that signature verification operators evaluate to a term that is already obvious from the term structure, hence we can assume that M' does not contain verification operations.

observation is that we can suitably pad, e.g., signatures such that the signing algorithm can be expressed as a length-dependent KDM circuit.) On the other hand, clique security, or even security against a polynomial number of arbitrary but predetermined KDM functions is *not* sufficient to treat such richer classes of calculi.

CCA security. We did not address security against *active* attacks in this paper. It seems plausible that one can use the technique of Camenisch et al. [10] to improve our bounded KDM secure scheme to security against chosen-ciphertext attacks (CCA security), using the techniques of Camenisch et al. [10]. ([10] use a modification of the Naor-Yung paradigm to improve the security of the circular secure encryption scheme of Boneh et al. [7] to CCA security.) Since CCA secure encryption suffices for computational soundness against active adversaries in the *absence* of key-cyclic expressions (e.g., [5, 26]), this could allow for full computational soundness against active adversaries. We see this as an interesting research direction.

7 Extending Haitner and Holenstein's Impossibility Result

In this section we observe that a result of Haitner and Holenstein [18], showing that there is no KDM-secure scheme with a proof of security which makes a black-box use of both the adversary and the KDM function, can be extended to rule out not just *full* KDM security but also *bounded* KDM security. The idea is simple: while this result used a *random* function h for the KDM function, a *pseudorandom* function could work just as well.
The following definition is adopted from [18].

Definition 7.1 (Cryptographic games)**.** A cryptographic game is a (possibly inefficient) random system Γ, where on security parameter k, Γ interacts with an attacker A and may output 1. We define the game value of such an interaction, denoted $\Gamma_A(1^k)$, as the probability that Γ outputs 1 in the end of the interaction with A, where the probability is taken over the random coins of Γ and A. A cryptographic game is non-interactive if it consists of two messages, from Γ to A and back.

Examples:

OWF. The security of a one-way function f is equivalent to requiring that the value of the following game is negligible for any efficient A. On security parameter k, the system Γ selects a random $x \in \{0,1\}^k$ and sends $y = f(x)$ to the adversary. Γ outputs 1 if A outputs $x' \in f^{-1}(y)$.

DDH. The security of the DDH hardness assumption is equivalent to requiring that the value of the following game is at most negligibly bounded from $\frac{1}{2}$ for any efficient adversary. Let G be an appropriate DDH group (e.g., Z_p^* for some prime p) and let g be a generator in the group. The system Γ chooses a random bit b, sends the tuple (g^x, g^y, g^z) to the challenger A, where x and y are random exponents, and $z = x \cdot y$ if $b = 0$ and a random value otherwise. Γ outputs 1 iff A has guessed b correctly.

Definition 7.2 (Strongly-black-box reductions)**.** An encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$ has a δ-strongly-black-box reduction from its KDM security to a cryptographic game Γ with respect to a family of query function $\mathcal{Q}$, if there exists an oracle-aided algorithm R with the following guarantee: Let A be an efficient adversary that breaks the KDM security of the scheme using query functions from $\mathcal{Q}$ with advantage $\epsilon_A = \epsilon_A(k)$ (i.e., On security parameter k, A distinguishes between encryptions

of functions of the secret key and encryptions of garbage with advantage $\epsilon_A(k)$). Then the value of $\Gamma_{R^A}(1^k) \geq \delta(k, \epsilon_A(k))$. Here, we require that R treat the query functions it gets from A as black boxes — all it can do is to query them on arbitrary chosen inputs.

Informally, we say that a proof for the KDM security of a scheme is strongly-black-box with respect to a game Γ and a family of query function $\mathcal{Q}$, if the value of $\delta(k, \epsilon_A(k))$ for every non-negligible $\epsilon_A(k)$ is considered a "break" of Γ (i.e., $\delta(k, \epsilon_A(k)) > \frac{1}{2} + \mathsf{negl}$ for the DDH game). We remark that *all* known KDM constructions in the literature have strongly-black-box reductions with respect to the relevant hardness assumption (e.g., DDH) and the class of query functions they are secure against.

Definition 7.3 (Pseudorandom functions (PRF))**.** An ensemble of functions $\mathcal{F} = \{\mathcal{F}_k = \{f\colon \{0,1\}^{m(k)} \mapsto \{0,1\}^{\ell(k)}\}\}$ is pseudorandom, if, on security parameter k, an efficient adversary cannot distinguish with more than negligible advantage between a random $f \in \mathcal{F}_k$, and a truly random function defined on the same input and output domains. Here, the adversary may only access the function as a black box. The ensemble is α-exponential hard for a constant $\alpha > 0$, if no adversary that runs in time 2^{n^α} wins in the above game with advantage greater than 2^{k^α}.

Theorem 7.4. *Let* $(\mathsf{Enc}, \mathsf{Dec})$ *be a* δ*-strongly-black-box reduction from its KDM security to a non-interactive cryptographic game* Γ *with respect to a family of query functions* $\mathcal{Q} = \{\mathcal{Q}_k\}$.[18] *Assume that* $\mathcal{Q}_k$ *contains the family of functions* $\mathcal{G}_k = \{g_k(x) = f(x, 0^{t(k)-k}) \oplus r\colon f \in \mathcal{F}_{t(k)}, r \in \{0,1\}^{\ell(k)}\}$*, where* $t(k) \geq 2k$ *and* $\mathcal{F}_{t(k)} = \{f\colon \{0,1\}^{t(k)} \mapsto \{0,1\}^{\ell(k)}\}$ *is an* α*-exponential hard PRF with* $t(k)^\alpha \geq 2k$*. Then, there exists an efficient algorithm* A *with* $\Gamma_A \geq \delta(k, 1 - 2^{-k}) - 2^{-k}$*.*

In particular, giving such a strongly-black-box reduction implies that either the class of query function considered is weak (does not contain exponentially hard PRF), or the game Γ can be efficiently broken with probability $\delta(k, 1 - 2^{-k}) - 2^{-k}$.

Proof (sketch) The proof is similar to the proof of [18, Theorem 5]. Consider the following (inefficient) adversary A for breaking the KDM security of $(\mathsf{Enc}, \mathsf{Dec})$ with respect to $\mathcal{Q}$. On security parameter k, choose a random $g \in \mathcal{G}_k$ and make a KDM query to obtain a ciphertext C. Then check (via exhaustive search) if there exists $sk \in \{0,1\}^k$ such that $\mathsf{Dec}_{sk}(C) = g(sk)$. If positive return 1, otherwise return 0. It is easy to verify that A breaks the KDM security with advantage $1 - 2^{-k}$ (the probability that a decryption of random ciphertext C equals to $g(sk)$ for some sk, is bounded by $\sum_{sk} \Pr[\mathsf{Dec}_{sk}(C) = g(sk)] = \sum_{sk} \Pr_r[\mathsf{Dec}_{sk}(C) = f(sk) \oplus r] \leq 2^k \cdot 2^{-2k} = 2^{-k}$). More interestingly, we notice that the probability that R^A sends a ciphertext $C = \mathsf{Enc}_{sk}(g(sk))$ to A without previously making the query $g(sk)$ is bounded by 2^{-2k}. Assume otherwise, then R^A is an algorithm that runs in time poly $\cdot 2^k$ and breaks the security of $\mathcal{F}$. It follows that we can emulate the execution of R^A: throughout the execution, keep track of all queries that R makes to g, and let T denote the list of queries. When R queries A on a ciphertext C, act as the inefficient A above, but *only* with respect to the secret keys in T. The above observation yields that we emulate R^A with error bounded by 2^{-2k}. ∎

[18] Theorem 7.4 can be shown to hold also against all natural interactive games (see [18] for details). For the sake of simplicity, however, we choose to focus here on the non-interactive case.

References

[1] M. Abadi and P. Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). *Journal of Cryptology*, 15(2):103–127, 2002.

[2] P. Adão, G. Bana, J. Herzog, and A. Scedrov. Soundness of formal encryption in the presence of key-cycles. In S. De Capitani di Vimercati, P. F. Syverson, and D. Gollmann, editors, *ESORICS 2005*, 2005.

[3] B. Applebaum, Y. Ishai, and E. Kushilevitz. Computationally private randomizing polynomials and their applications. *Computational Complexity*, 15(2):115–162, 2006.

[4] B. Applebaum, D. Cash, C. Peikert, and A. Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In S. Halevi, editor, *CRYPTO 2009*, 2009.

[5] M. Backes, B. Pfitzmann, and M. Waidner. A composable cryptographic library with nested operations (extended abstract). In *CCS 2003*, pages 220–230, 2003. Full version in IACR Cryptology ePrint Archive 2003/015, Jan. 2003, `http://eprint.iacr.org/`.

[6] J. Black, P. Rogaway, and T. Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In K. Nyberg and H. M. Heys, editors, *SAC 2002*, 2003.

[7] D. Boneh, S. Halevi, M. Hamburg, and R. Ostrovsky. Circular-secure encryption from decision diffie-hellman. In D. Wagner, editor, *CRYPTO 2008*, 2008.

[8] Z. Brakerski, S. Goldwasser, and Y. Kalai. Circular-secure encryption beyond affine functions. Cryptology ePrint Archive, Report 2009/485, 2009. `http://eprint.iacr.org/`.

[9] J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In B. Pfitzmann, editor, *EUROCRYPT 2001*, 2001.

[10] J. Camenisch, N. Chandran, and V. Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In A. Joux, editor, *EUROCRYPT 2009*, 2009.

[11] C. Crépeau and J. Kilian. Weakening security assumptions and oblivious transfer. In S. Goldwasser, editor, *CRYPTO 1988*, 1990.

[12] I. Damgård and M. J. Jurik. A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In *PKC*, 2001.

[13] D. Dolev and A. C. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, 1983.

[14] S. Even and O. Goldreich. On the security of multi-party ping-pong protocols. In *FOCS 1983*, 1983.

[15] S. Even, O. Goldreich, and A. Lempel. A randomized protocol for signing contracts. *Commun. ACM*, 1985.

[16] C. Gentry. Fully homomorphic encryption using ideal lattices. In M. Mitzenmacher, editor, *STOC*, 2009.

[17] O. Goldreich and L. A. Levin. A hard-core predicate for all one-way functions. In *STOC*, 1989.

[18] I. Haitner and T. Holenstein. On the (im)possibility of key dependent encryption. In O. Reingold, editor, *TCC 2009*, 2009.

[19] S. Halevi and H. Krawczyk. Security under key-dependent inputs. In P. Ning, S. D. C. di Vimercati, and P. F. Syverson, editors, *ACM CCS 07*, 2007.

[20] D. Hofheinz and D. Unruh. Towards key-dependent message security in the standard model. In N. Smart, editor, *EUROCRYPT 2008*, 2008.

[21] Y. Ishai and A. Paskin. Evaluating branching programs on encrypted data. In *TCC 2007*, 2007.

[22] R. Kemmerer, C. Meadows, and J. Millen. Three systems for cryptographic protocol analysis. *Journal of Cryptology*, 7(2):79–130, 1994.

[23] Y. Lindell and B. Pinkas. A proof of security of Yao's protocol for two-party computation. *J. Cryptology*, 22(2):161–188, 2009. Earlier version in ECCC TR 2004-063.

[24] G. Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In *TACAS 1996*, 1996.

[25] M. Merritt. *Cryptographic Protocols*. PhD thesis, Georgia Institute of Technology, 1983.

[26] D. Micciancio and B. Warinschi. Soundness of formal encryption in the presence of active adversaries.

In *TCC 2004*, 2004.
[27] M. Naor. On cryptographic assumptions and challenges. In *CRYPTO 2003*, 2003.
[28] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT 1999*, 1999.
[29] M. Rabin. How to exchange secrets by oblivious transfer. Technical Report TR-81, Harvard Aiken Computation Laboratory, 1981.
[30] A. C. Yao. How to generate and exchange secrets. In *FOCS 1986*, 1986.

A The BHHO Scheme

In this section we briefly survey the expanded scheme of Boneh et al. [7] and explain how we can use it to obtain (augmented) targeted encryption.

Theorem A.1 (Implicit in [7])**.** *Under the DDH assumption, there is a targeted encryption protocol* $\mathsf{TES} = (\mathsf{TGen}, \mathsf{TEnc}, \mathsf{TDec})$*. Furthermore, it is possible to augment* TES *with encryption and decryption algorithms so that* $\mathsf{ATES} = (\mathsf{TGen}, \mathsf{TEnc}, \mathsf{TDec}, \mathsf{Enc}, \mathsf{Dec})$ *is a circular-secure augmented targeted encryption protocol.*

sketch We now sketch how the "expanded scheme" of [7] (henceforth BHHO) implies such a tuple of algorithms. To do so, it will be useful to follow [7] (albeit with somewhat different notational conventions) and use *additive notation* for the group they are working on.

The BHHO scheme works in some group $\mathbb{G}$ that is isomorphic to $\mathbb{Z}_q$ for some q. (Specifically, the group will be the multiplicative group generated by some element g of order q modulo p for some prime p of the form $c \cdot q + 1$.) Of course this isomorphism is not efficiently computable, but still it turns out to be convenient to describe all operations as if they happen in $\mathbb{Z}_q$ and later verify that they can indeed be carried out efficiently. The three algorithms are defined as follows:

Key Generation. $\mathsf{TGen}(1^k)$ chooses random $s \in \{0,1\}^\ell \subseteq \mathbb{Z}_q^\ell$, the secret sk key is the $\ell + 1$ dimensional vector $(s, 1)$. The public key pk is a random $(\ell + 1) \times \ell$ dimensional matrix A of rank ℓ such that the image of A is the space orthogonal to $(s, 1)$. (The exact way that A is chosen, and the exact distribution of A are immaterial for our purposes, as long as this is done efficiently.)

Targeted encryption. $\mathsf{TEnc}_{pk,i,b}(x)$ for message $x \in \mathbb{Z}_q$ with index i and bit b chooses a random column vector $r \in \mathbb{Z}_q^\ell$ and proceeds as follows:

- If $b = 1$ then output $Ar + xe_i$ where e_i is the $\ell + 1$ dimensional column vector that has 1 in the i^{th} coordinate and 0 everywhere else.
- If $b = 0$ then output $Ar - xe_i + xe_{\ell+1}$.

Decryption. $\mathsf{TDec}_{sk}(w)$ decrypts a vector $w \in \mathbb{Z}_q^{\ell+1}$ by just taking its inner product with the secret key $sk = (s, 1)$.

We now sketch the three properties that need to be satisfied for targeted encryption:

Targeted decryption. The inner product of Ar and sk is zero, so the Ar term will not affect the output of the decryption algorithm. Now if $s_i = 1$ then the inner product of xe_i and $(s, 1)$ will be x. If $s_i = 0$ then the inner product of $-xe_i$ and $(s, 1)$ is 0 and the inner product of $xe_{\ell+1}$ with $(s, 1)$ is x. In both cases we get that $\mathsf{Dec}_{sk}(\mathsf{Enc}_{pk,i,sk_i}(x)) = x$.

Security against receiver. If $s_i = 0$, then $Ar + xe_i$ is a random vector in the space orthogonal to $(s, 1)$ regardless of what x is. Similarly if $s_i = 1$ then $Ar + xe_i - xe_{\ell+1}$ is a random vector in this space as well.

Security against outsiders. In the case $s_i = b$, the output of $\mathsf{Enc}_{pk,i,b}(x)$ is a random vector in the affine space of vectors w such that their inner product with sk is x. [7] show that under DDH this is indistinguishable from random vectors in the space orthogonal to sk, even to distinguishers that have the public key.

In the actual scheme, the public key is not represented as a matrix in $\mathbb{Z}_q$ but rather, using exponentiation, as a matrix of elements in $\mathbb{G}$. Still, one can easily verify that it's possible to efficiently carry out inner products and matrix products as long as that at least one of the vectors is given in "un-exponentiated form" (i.e. with entries in $\mathbb{Z}_q$ and not in $\mathbb{G}$). We will consider the message x as an element in $\mathbb{G}$, and so the goal of the decryption algorithm is not to recover the discrete log of x, but just x itself.

Finally, we can augment $\mathsf{TES} = (\mathsf{TGen}, \mathsf{TEnc}, \mathsf{TDec})$ with the original encryption and decryption algorithms from [7]. Since their scheme is circular-secure, this yields a circular-secure augmented targeted encryption protocol $\mathsf{ATES} = (\mathsf{TGen}, \mathsf{TEnc}, \mathsf{TDec}, \mathsf{Enc}, \mathsf{Dec})$. ∎

B The ACPS Scheme

From an abstract point of view, the LWE-based scheme of Applebaum et al. [4] is similar to the DDH-based scheme of Boneh et al. [7]. Hence [4] implies targeted encryption in a similar way to [7].

Theorem B.1 (Implicit in [4])**.** *Under the LWE assumption, there is a targeted encryption protocol* $\mathsf{TES} = (\mathsf{TGen}, \mathsf{TEnc}, \mathsf{TDec})$*. Furthermore, it is possible to augment* TES *with encryption and decryption algorithms so that* $\mathsf{ATES} = (\mathsf{TGen}, \mathsf{TEnc}, \mathsf{TDec}, \mathsf{Enc}, \mathsf{Dec})$ *is a circular-secure augmented targeted encryption protocol.*

Proof sketch The scheme of [4] (henceforth ACPS) operates in the ring $\mathbb{Z}_q$ for $q = p^2$, where p is a suitable *polynomial* in the security parameter k. Note that we can view $\mathbb{Z}_p$ as an additive subgroup of $\mathbb{Z}_q$. Furthermore, the scheme fixes a discretized Gaussian distribution χ over $\mathbb{Z}_q$, and parameters $n, m \in \mathbb{N}$ with $m > n$. The secret key of the ACPS scheme is an element $\mathbf{s} \in \mathbb{Z}_q^n$. We first show how to obtain a variant of targeted encryption, for which $\mathsf{TEnc}_{pk,i,s}(z)$, for $i \in [n]$, $s \in \mathbb{Z}_q$, and $z \in \mathbb{Z}_p$, decrypts to z iff $\mathbf{s}_i = s$, and contains (almost) no information about z for $\mathbf{s}_i \neq s$. (Hence, the difference to our original definition is that we encrypt values depending on secret key *elements* instead of secret key bits.)

Key Generation. $\mathsf{TGen}(1^k)$ samples $\mathbf{s} = (\mathbf{s}_1, \ldots, \mathbf{s}_n) \leftarrow \chi^n$ and uniform $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, and sets $\mathbf{b} := \mathbf{A}^T s + x$ for an independent error vector $x \leftarrow \chi^m$. Public key is $pk = (\mathbf{A}, \mathbf{b})$, and secret key is $sk = \mathbf{s}$.

Targeted encryption. $\mathsf{TEnc}_{pk,i,s}(z)$ for message $z \in \mathbb{Z}_p$, index $i \in [n]$, and $s \in \mathbb{Z}_q$ first chooses random $(\mathbf{u}', v) \in \mathbb{Z}_q^m \times \mathbb{Z}_q$ such that $v - \langle \mathbf{u}'; \mathbf{s} \rangle \in \mathbb{Z}_q$ is small with overwhelming probability. (Such an $(\mathbf{u}', v)$ can be chosen of the form $(\mathbf{Ar}, \langle \mathbf{r}; \mathbf{b} \rangle + e)$ for suitably small $\mathbf{r} \in \mathbb{Z}_q^m$ and $e \in \mathbb{Z}_q$. This is done as in a regular encryption, and the details are not important for our case.) Additionally, TEnc chooses uniformly $y \leftarrow \mathbb{Z}_q$ and outputs

$$C := (\mathbf{u}, c) := (\mathbf{u}' + p \cdot y\mathbf{e}_i, v + p \cdot (z + ys))$$

for the vector $\mathbf{e}_i \in \mathbb{Z}_q^m$ that has 1 in the i^{th} coordinate and 0 everywhere else.

Decryption. $\mathsf{TDec}_{sk}(\mathbf{u}, c)$ decrypts $(\mathbf{u}, c) \in \mathbb{Z}_q^m \times \mathbb{Z}_q$ to the $z \in \mathbb{Z}_p$ for which $p \cdot z \in \mathbb{Z}_q$ is closest to $c - \langle \mathbf{u}; \mathbf{s} \rangle \in \mathbb{Z}_q$.

We now sketch the three properties that need to be satisfied for targeted encryption:

Targeted decryption. Decryption of a ciphertext $(\mathbf{u}, c) \leftarrow \mathsf{TEnc}_{pk,i,s}(z)$ computes

$$c - \langle \mathbf{u}; \mathbf{s} \rangle = p \cdot (z + ys - \langle y\mathbf{e}_i; \mathbf{s} \rangle) + \left(v - \langle \mathbf{u}'; \mathbf{s} \rangle\right) = p \cdot (z + y(s - \mathbf{s}_i)) + \left(v - \langle \mathbf{u}'; \mathbf{s} \rangle\right). \quad (1)$$

By construction, $v - \langle \mathbf{u}'; \mathbf{s} \rangle$ is small, and hence decryption outputs $z + y(s - \mathbf{s}_i)$. In particular, $\mathsf{Dec}_{sk}(\mathsf{Enc}_{pk,i,\mathbf{s}_i}(z)) = z$.

Security against receiver. If $s \neq \mathbf{s}_i$, then the uniform y statistically blinds the decryption process in (1). Formally, [4, Lemma 5] implies that a ciphertext as constructed by $\mathsf{Enc}_{pk,i,s}(z)$ with $s \neq \mathbf{s}_i$ is statistically close to a uniform ciphertext of a uniform message.

Security against outsiders. $\mathsf{Enc}_{pk,i,\mathbf{s}_i}(z)$ is statistically close to an encryption of z. Hence, the semantic security of the encryption scheme of [4] implies security against outsiders under the LWE assumption. (Formally, TEnc can be formulated to only use a KDM encryption oracle for affine functions and the encryption scheme of [4]. [4] prove that such encryptions are computationally indistinguishable from fake encryptions under the LWE assumption.)

A targeted encryption protocol in the sense of Section 3 (in which encryptions depend on the individual *bits* of sk) can be constructed as follows. To encrypt a value z in a way such that it can only by retrieved if the j^{th} bit of $\mathbf{s}_i$ equals $b \in \{0,1\}$, we construct p ciphertexts $C_s \leftarrow \mathsf{TEnc}_{pk,i,s}(z_s)$ ($s \in \mathbb{Z}_p$), where $z_s = z$ iff the j^{th} bit of s equals b, and $z_s = 0$ else. (Recall that p is polynomial in the security parameter.)

Again, we can augment $\mathsf{TES} = (\mathsf{TGen}, \mathsf{TEnc}, \mathsf{TDec})$ with the original encryption and decryption algorithms from [4] to obtain a circular-secure augmented targeted encryption protocol. ∎

C Garbled Circuits

In this section we prove Theorem 2.1 by relying on the version of Yao's garbled circuit construction [30] which was analyzed in [3].[19]

Lemma C.1 ([3], Construction 4.7)**.** *Suppose that one-way functions exist. Then, for any polynomial-time computable function $f : \{0,1\}^k \to \{0,1\}^{m(k)}$ there is a pair of polynomial-time randomized algorithms* $(\mathsf{Encode}, \mathsf{Decode})$ *that for security/input parameter k satisfy the following:*

Syntax. Encode *takes a $2k$ key tuple $\overline{W} = \{W_{i,b}\}_{i \in [k], b \in \{0,1\}}$, where $W_{i,b} \in \{0,1\}^k$, and a k-bit input mask r, and outputs an "encoding" Z. (Intuitively, for any input u, the encoding Z together with $u \oplus r$ and the k keys corresponding to u can be used to recover $f(u)$ but reveal no additional information about u.)* Decode *takes a k-tuple of k-bit keys, a masked input of length k and an encoding Z, and outputs $y \in \{0,1\}^{m(k)}$. (Note that, in contrast to Theorem 2.1, here the input for f is not given to* Decode*.)*

[19] Several proofs of security of *protocols* which rely on the garbled circuit construction appear in the literature. In particular, a proof for a statistically correct variant of Theorem 2.1 can be obtained from the security proof of Yao's original protocol appearing in [23]. The analysis of the garbled circuit construction from [3] has the advantages that it supports perfect correctness and it can be used in a more modular way to derive Theorem 2.1.

Correctness. *For every input $u \in \{0,1\}^k$ for f, k key pairs $\overline{W}$, and input mask r, if $Z = \mathsf{Encode}(\overline{W}, r)$ then $\mathsf{Decode}(\overline{W}_u, u \oplus r, Z) = f(u)$, where here we define $\overline{W}_u = \{W_{i,u_i}\}_{i\in[k]}$ (without including u in $\overline{W}_u$).*

Privacy. *For every $u, u' \in \{0,1\}^k$ such that $f(u) = f(u')$, if $\overline{W}$ and r are chosen at random then*

$$(\overline{W}_u, u \oplus r, \mathsf{Encode}(\overline{W}, r)) \stackrel{c}{\approx} (\overline{W}_{u'}, u' \oplus r, \mathsf{Encode}(\overline{W}, r)).$$

Given $(\mathsf{Encode}, \mathsf{Decode})$ as in Lemma C.1 we construct $(\mathsf{Garble}, \mathsf{GCEval})$ as required by Theorem 2.1. We start by describing a construction which does not consider security against outsiders, and later describe a simple generic transformation which guarantees this additional property.

Construction C.2. Let U denote a universal function with the following inputs and output. The input u of U includes a description h of a circuit of size s computing a function $h : \{0,1\}^k \to \{0,1\}^m$ along with an input $x \in \{0,1\}^k$ for h. The output of U is $h(x)$. We may assume without loss of generality that the description size $|h|$ of h is fully determined by k and s and that $h \circ x$ can be uniquely parsed as (h, x).

Let $(\mathsf{Encode}, \mathsf{Decode})$ be as promised by Lemma C.1 for $f = U$. We define $(\mathsf{Garble}, \mathsf{GCEval})$ as follows:

- Garble, given k pairs of k-bit keys $\overline{K} = \{K_{i,b}\}$ and a circuit $h : \{0,1\}^k \to \{0,1\}^m$ of size s, picks $|h|$ additional pairs of k-bit keys $\overline{H} = \{H_{j,b}\}$ and random input masks $r' \in \{0,1\}^{|h|}$ and $r'' \in \{0,1\}^k$. Letting $\overline{W} = \overline{H} \circ \overline{K}$ and $r = r' \circ r''$, Garble invokes $\mathsf{Encode}(\overline{W}, r)$ and obtains an output Z. It lets $GC = (\overline{H}_h, h \oplus r', r'', Z)$.
- GCEval, given x, a k-tuple of keys $\overline{K}_x$ and a garbled circuit $GC = (\overline{H}_h, h \oplus r', r'', Z)$, lets $\overline{W}_u = \overline{H}_h \circ \overline{K}_x$ and invokes $\mathsf{Decode}(\overline{W}_u, (h \oplus r') \circ (x \oplus r''), Z)$.

Claim C.3. *If $(\mathsf{Encode}, \mathsf{Decode})$ satisfy the requirements of Lemma C.1 then $(\mathsf{Garble}, \mathsf{GCEval})$ defined in Construction C.2 satisfy the correctness and security against receiver requirements from Theorem 2.1.*

Proof **Correctness:** Since Z is an output of $\mathsf{Encode}(\overline{W}, r)$, the correctness property of Decode guarantees that $\mathsf{Decode}(\overline{W}_u, (h \oplus r') \circ (x \oplus r''), Z) = \mathsf{Decode}(\overline{W}_{h\circ x}, (h \circ x) \oplus r, Z) = U(h, x) = h(x)$ as required.

Security against receiver: Let $u = (h, x)$ and $u' = (h', x)$ such that $|h| = |h'|$ and $h(x) = h'(x)$ (so that $U(u) = U(u')$). The privacy of Encode guarantees that for a random choice of $\overline{W} = \overline{H} \circ \overline{K}$ and $r = (r', r'')$,

$$(\overline{W}_u, (h \oplus r') \circ (x \oplus r''), Z) \stackrel{c}{\approx} (\overline{W}_{u'}, (h' \oplus r') \circ (x \oplus r''), Z),$$

where $Z = \mathsf{Encode}(\overline{W}, r)$. It follows that

$$(\overline{W}_u, (h \oplus r') \circ r'', Z) \stackrel{c}{\approx} (\overline{W}_{u'}, (h' \oplus r') \circ r'', Z),$$

which implies that

$$\overline{K}_x \circ \mathsf{Garble}(\overline{K}, h) \equiv (\overline{W}_u, (h \oplus r') \circ r'', Z) \stackrel{c}{\approx} (\overline{W}_{u'}, (h' \oplus r') \circ r'', Z) \equiv \overline{K}_x \circ \mathsf{Garble}(\overline{K}, h')$$

as required. ∎

The abstract properties of Encode alone do not guarantee that Construction C.2 provides security against outsiders. However, it is easy to enforce this extra requirement in a generic way by using a portion of the keys $K_{1,0}, K_{1,1}$ to "encrypt" the garbled circuit. Concretely, augment $(\mathsf{Garble}, \mathsf{GCEval})$ to use only the first $k/2$ bits of each k-bit key and let the new garbled circuit GC consist of a pair of encryptions of the output of Garble: one with the second half of $K_{1,0}$ and one with the second half of $K_{1,1}$. This makes GC useless to an outsider, while respecting the correctness and security against receiver property of $(\mathsf{Garble}, \mathsf{GCEval})$.