On the (Im)Possibility of Key Dependent Encryption

Iftach Haitner* Thomas Holenstein†

Abstract

We study the possibility of constructing encryption schemes secure under messages that are chosen depending on the key k of the encryption scheme itself. We give the following separation results that hold both in the private and in the public key settings:

- Let $\mathcal{H}$ be the family of poly(n)-wise independent hash-functions. There exists no fully-black-box reduction from an encryption scheme secure against key-dependent messages to one-way permutations (and also to families of trapdoor permutations) if the adversary can obtain encryptions of $h(k)$ for $h \in \mathcal{H}$.
- There exists no reduction from an encryption scheme secure against key-dependent messages to, essentially, *any* cryptographic assumption, if the adversary can obtain an encryption of $g(k)$ for an *arbitrary* g, as long as the reduction's proof of security treats both the adversary and the function g as black boxes.

Keywords: Key-dependent input, Black-box separations, One-way functions

1 Introduction

A cryptographic primitive is *key-dependent input secure*, or KDI-secure for short, if it remains secure also in case where the input depends on the secret key. In the case of encryption schemes, KDI-security means that the adversary can obtain, in addition to the usual queries, encryptions of $h(k)$, where k is the key of the scheme and h is chosen by the adversary from some (hopefully large) family of functions $\mathcal{H}$.

On a first look it might seem that by using the right design it is possible to prevent any "KDI-attacks" on the encryption scheme, and thus achieving such strong security would be only of pure theoretical interest. It turns out, however, that such attacks might "naturally" arise when considering complex systems. For instance, in the BitLocker disk encryption utility (used in Windows Vista), the disk encryption key can end up being stored on the disk and thus encrypted along with the disk contents. For more details on the importance of KDI-security, see [BHHO08] and references within.

In this work we study the possibility of obtaining such an encryption scheme both from one-way functions and from other hardness assumptions. In particular, we exclude different types of reductions from a KDI-secure encryption scheme to different hardness assumption. Intuitively, a *black-box reduction* of a primitive P to a primitive Q is a construction of P out of Q that ignores the internal structure of the implementation of Q and just uses it as a "subroutine" (i.e., as a black box). The reduction is *fully-black-box* (following [RTV04]) if the proof of security (showing that an adversary that breaks the implementation of P implies an adversary that breaks the implementation of Q) is also black-box (i.e., the internal structure of the adversary that breaks the implementation of P is ignored as well).

Our first result shows that there is no fully-black-box reduction from KDI-secure encryption schemes to one-way permutations, even if the KDI-security is only against the relatively small class of poly(n)-wise independent hash-functions. When considering reduction from a KDI-secure encryption scheme, it is

*Microsoft Research, New England Campus. E-mail: iftach@microsoft.com. This work was performed while at Weizmann Institute of Science and at Microsoft Research, Silicon Valley Campus.

†Department of Computer Science, Princeton University. E-mail: tholenst@princeton.edu. This work was performed while at Microsoft Research, Silicon Valley Campus.

natural to ask whether the proof of security accesses the challenge function h in a black-box manner as well. Our second result, however, shows that under this restriction essentially no hardness assumption implies a KDI-secure encryption scheme.

1.1 Related Work

KDI security. The development of encryption secure against key-dependent inputs started by the works of Abadi and Rogaway [AR02]. They studied formal security proofs for cryptographic protocols (as described by [DY83]), and showed that these imply security by a reduction, as long as no *key-cycles* exist in the protocol, i.e., there is a partial order $\preceq$ on the keys exists such that a message depending on k_1 would only be encrypted with k_2 if $k_1 \preceq k_2$. Since this is a restriction (even though it may be a very natural one), the community became aware that it would be desirable to create encryption schemes that provide security even in the existence of such key cycles. Consequently, Black, Rogaway, and Shrimpton [BRS02] define the (possibly stronger) notion of KDI-security for symmetric encryption schemes, and show how to obtain this notion in the random-oracle model. In such a scheme, an adversary can obtain encryptions of $h(k)$ under the key k, where h is given as a circuit to an encryption oracle. Such a scheme implies the security of the scheme under cycles as well. Independently of [BRS02], a notion of circular security has been defined by Camenisch and Lysyanskaya [CL01], considering asymmetric encryption schemes as well.

Recently, Halevi and Krawczyk [HK07] generalized the notion of KDI-security to other cryptographic primitives, such as pseudorandom functions. They also coined the name KDI for this sort of security (previously, it was named *key-dependent message security*). Their results in this setting are mainly for the construction of pseudorandom functions. In addition, [HK07] shows that a *deterministic* encryption scheme cannot be KDI-secure. Independently and concurrently of [HK07], Hofheinz and Unruh [HU08] provided private-key encryption schemes secure under a limited class of KDI-attacks. The main limitation of their work is that the scheme only remains secure as long as $h(k)$ is significantly shorter than the key; also, after every application of the encryption scheme the key is updated. This makes the construction insufficient for the initial motivation of allowing key cycles in cryptographic protocols. Very recently, Boneh et al. [BHHO08] presented a public-key encryption scheme that is KDI-secure (assuming that the DDH assumption holds) against the family of affine transformations over the messages' domain. Their system remains secure also when key-cycles are allowed.

Black-box impossibility results. Impagliazzo and Rudich [IR89] showed that there are no black-box reductions of key-agrement protocols to one-way permutations and substantial additional work in this line followed (cf., [GKM$^+$00, Rud88, Sim98]). Kim, Simon and Tetali [KST99] initiated a new line of impossibility results, by providing a lower bound on the *efficiency* any black-box reduction of universal one-way hash functions to one-way permutations, substantial additional work in this line followed (cf., [GGKT05, HHRS07, HK05, Wee07]). Dodis et al. [DOP05] (and also [Hof08]) give a black-box separation of a similar flavor to the one given in Theorem 1.2, in the sense that it excludes a large family of hardness assumptions.

1.2 Contributions of this Paper

In this paper we give two impossibility results for security proofs of constructions of KDI-secure private-key encryption schemes. However, since every public-key encryption scheme can be also viewed as a private-key scheme (i.e., both parties use the same private/public key), our impossibility results immediately extend to the public-key case. Our first result is a black-box separation of KDI encryption scheme from one-way permutations and from (even enhanced) family of trapdoor permutations.

Theorem 1.1. *Let* $(\mathrm{Enc}, \mathrm{Dec})$ *be an encryption scheme that is fully-black-box constructed from one-way permutations. Then there exists an efficient family* $\mathcal{H}$ *of* $\mathrm{poly}(n)$*-wise independent hash functions such that the following holds: there exists no black-box reduction from breaking the KDI-security of* $(\mathrm{Enc}, \mathrm{Dec})$ *against* $\mathcal{H}$ *to inverting one-way permutations. Furthermore, the above holds also with respect to (enhanced) families of trapdoor permutations.*

For our second result, we assume that the challenge function itself under which the scheme should be KDI-secure is treated by the proof of security as a black box. Moreover, the proof of security does not forward an access to the challenge function to a "third party". We call such a reduction ***strongly-black-box*** and give the following result.

Theorem 1.2 (informal)**.** *There exists no reduction with strongly-black-box proof from the KDI-security of an encryption scheme to the security of "any cryptographic assumption".*

We stress that the construction of the encryption scheme considered in Theorem 1.2 can be arbitrary. The formal statement of Theorem 1.2 is given in Section 4.

1.3 Interpretation of Our Results

So what should we think on the possibility of building KDI-secure encryption scheme given the above negative results? Let us start with Theorem 1.1 and let's first consider the fully-black-box restrictions. We remark that while quite a few black-box impossibility results of these types are known (see Section 1.1), there is not even a single known example where we have an impossibility result of the type given in Theorem 1.1, and yet a "non-black-box" reduction was found.[1] We also remark that the reductions given in [BHHO08, HK07, HU08] are fully-black-box. The second issue is that we only rule out security against poly-wise independent hash function, where the value for "poly" is determined as a function of the encryption scheme. It seems, however, that in most settings one cannot limit the power of the queries used in the KDI attack (but merely assume that these functions should be efficiently computable). Typically, when designing an encryption scheme, the exact configuration of each of the systems in which the scheme is going to be used is unknown. These configurations, however, determine the challenge functions "used" in the KDI attacks.

In Theorem 1.2 we consider arbitrary constructions, but require only black-box access to the challenge functions. This additional restriction actually reflects three separate restrictions. The first is that the proof has only input/output access to the challenge function, the second is that the challenge function cannot be assumed to be efficient, and the third is that the reduction "knows" all the queries made to the challenge functions (we force the last restriction by disallowing the reduction to give a third party an handle to the challenge function). While the first two restrictions seem to be a real limitation on the generality of our second result, the third restriction is harmless in most settings. In particular, this is the case where the hardness assumption does not accept (even implicitly) handler to functions. This list includes all the "non-interactive hardness assumptions" such as one-way functions, factoring, DDH etc.[2]

1.4 Our Technique

In the proof of both our results, we are using the same oracle, Breaker, that helps us to break the KDI-security of every encryption scheme. Let $(\mathrm{Enc}, \mathrm{Dec})$ be some fixed encryption scheme. On input (h, c), where h is some *length doubling* function and c is a ciphertext, Breaker considers all possible keys, and returns the first key k for which $\mathrm{Dec}(k, c) = h(k)$, or $\perp$ if no such key exists. It is not hard to see that $(\mathrm{Enc}, \mathrm{Dec})$ is not KDI-secure, with respect to h, in the presence of Breaker. Therefore, our impossibly results follow if Breaker does not help to violate the underlying hardness assumption. For this, we need to assume that Breaker is only called with functions h that are chosen uniformly from the respective set of challenge functions. We ensure this by restricting the functions h for which Breaker performs the above computation to one that is randomly chosen (and then give the adversary access to it). Under this restriction, we manage to prove that Breaker does not help in breaking the assumption. Proving this is our main technical contribution (note that Breaker cannot be implemented efficiently) and we prove it differently in each of our separation results.

[1] The superiority of non-black-box techniques was demonstrated by Barak [Bar01] in the settings of zero-knowledge arguments for NP. In these settings, however, the black-box access is to the, possibly cheating, verifier and not to any underlying primitive.

[2] The only exception we could think of for a reduction that benefits from passing the handler to the challenge function to a third party, is a reduction from one KDI-secure encryption scheme B to another KDI-secure encryption scheme A. In such a reduction, the security proof of scheme B typically forwards the challenge function to the security proof of scheme A.

One-way permutations. Let π be a random permutation and let $(\mathrm{Dec}^{\pi}, \mathrm{Enc}^{\pi})$ be a candidate of a KDI-secure encryption scheme given π as the one-way permutation. We find a polynomial $p(n)$ (which depends on Dec and Enc) and use a family of length-doubling $p(n)$-wise independent hash-functions as the challenge functions. Imagine now that a call of A to Breaker helps A to invert π. Then, the behavior of Breaker must be very different for a large number of potential preimages of y, as otherwise the call gave roughly no information about the preimage of y. We show, however, that for all but a negligible fraction of the functions h, the behavior of Breaker will be the same for most possible preimages of y, no matter how the ciphertext is chosen.

Arbitrary assumptions. In this we use the family of *all* length-doubling functions as the challenge functions. The idea is that for a random h in the family, all calls to Breaker (done outside of the KDI game) are very likely to be answered with $\perp$. The reason is that for a fixed $k \in \{0,1\}^t$, the probability that $\mathrm{Dec}(k, c) = h(k)$ is roughly 2^{-2t}. This somewhat naive intuition is actually false, as it can fail in the following way: A picks a key k' itself, queries $h(k')$, encrypt this with k' itself, and gives the resulting ciphertext to Breaker. We prove, however, that this is essentially the only way in which the above intuition fails. Thus, instead of calling Breaker, A can as well check the keys on which h was previously queried, which can be done efficiently. We conclude that if there is a reduction with strongly-black-box proof of security from a KDI encryption scheme to a given hardness assumption, then the hardness assumption is false.

1.5 Paper Organization

We present the notations and formal definitions used in this paper in Section 2. In Section 3 we give the separation from one-way permutations, and in Section 4 we give the separation from any hardness assumption. Finally, in Section 5 we consider the question whether the techniques used in this paper are also useful for proving impossibility result w.r.t. other KDI-secure cryptographic primitives.

2 Preliminaries

2.1 Notation

We denote the concatenation of two strings x and y by $x \circ y$. If X is a random variable taking values in a finite set $\mathcal{U}$, then we write $x \leftarrow \mathcal{U}$ to indicate that x is selected according to the uniform distribution over $\mathcal{U}$. We often use probabilities where we choose an oracle π from some uncountable set of oracles at random. To make these well defined, one has to define the uniform measure on the oracles, which can be done by mapping an oracle to a real $r \in [0,1)$ (for example by interpreting the truth table of the oracle as binary representation of r), and then using Lebesgue-measure. One can verify that with this definition indeed all random variables that occur in this paper are measurable.

2.2 Many-wise Independence

Definition 2.1 (s-wise independent random variables)**.** *Let $X_1, \ldots, X_n$ be random variables defined over $\mathcal{U}$. $X_1, \ldots, X_n$ are s-wise independent for some $s \in N$ if every sequence $(X_{i_1}, \ldots, X_{i_s})$ of random variables is uniformly distributed over $\mathcal{U}^s$.*

Definition 2.2 (s-wise independent hash functions)**.** *Let $\mathcal{H}$ be a family of functions mapping $\{0,1\}^{\ell}$ to $\{0,1\}^m$, $s \in \mathbb{N}$. We say that $\mathcal{H}$ is an* efficient family of s-wise independent hash functions *(following [CW79]) if the following hold:*[3]

Samplable. *$\mathcal{H}$ is samplable in time* $\mathrm{poly}(s, \ell, m)$.

[3]The first two properties, regarding the efficiency of the family, implicitly assume an ensemble of families (one family for every triple (s, ℓ, m)). For simplify of presentation, we only refer to a single family.

Efficient. *There exists an algorithm that given $x \in \{0,1\}^\ell$ and a description of $h \in \mathcal{H}$ outputs $h(x)$ in time* $\mathrm{poly}(s, \ell, m)$.

s-wise independence. *If h is chosen at random from $\mathcal{H}$, the 2^ℓ random variables $h(x)$, $x \in \{0,1\}^\ell$ are s-wise independent.*

It is well known ([CW79]) that there exists an efficient family of s-wise independent hash functions from ℓ-bit strings to m-bit strings for every choice of ℓ and m whose elements description size is $\mathcal{O}(s \cdot \max\{\ell, m\})$.

We use the following upper bound on the probability that many s-wise independent events occur, where each event has low probability. The usual bounds seem not strong enough in our setting, as they focus on the range where the probability of a single event is constant. In our case, the probability of a single event decreases as the number of events increases, and we therefore use a different bound.

Lemma 2.3. *For $s, V \in \mathbb{N}$ let $B_1, \ldots, B_V$ be s-wise independent Bernoulli random variables with $\mathsf{Pr}[B_i = 1] \leq \frac{1}{V}$. Assuming that $\alpha > s$, then $\mathsf{Pr}\left[\sum_{i=1}^{V} B_j \geq \alpha\right] < \frac{\log(V)}{\alpha^{s-1}}$.*

Proof. The idea of the proof is to find a small family, $\mathcal{T}$, of sets of size s in $\mathcal{U} = \{1, \ldots, V\}$, such that every set in $\mathcal{U}$ of size α has a subset in $\mathcal{T}$. In this case, one can use the union bound on the sets in $\mathcal{T}$ (as explained in the proof). We first find $\mathcal{T}$.

Claim 2.4. *There exists a family of $\alpha \log(V)(\frac{V}{\alpha})^s$ sets of size s such that the following holds. Every set $Q \subseteq \mathcal{U}$ of size $|Q| = \alpha$ has a subset $\tau \subseteq Q$ in $\mathcal{T}$.*

Proof. Directly with the probabilistic method. Pick $\alpha \log(V)(\frac{V}{\alpha})^s$ sets of size s at random. The probability that a fixed set of size α has none of these sets as a subset is $(1 - (\frac{\alpha}{V})^s)^{\alpha \log(V)(\frac{V}{\alpha})^s} < e^{-\alpha \log(V)}$. Since there are $\binom{V}{\alpha} < V^\alpha = e^{\alpha \log(V)}$ subsets of size α in $\mathcal{U}$, the expected number of uncovered sets is smaller than 1, which implies that with positive probability we obtain a family as required. □

Consider now a family $\mathcal{T}$ with $|\mathcal{T}| = \alpha \log(V)(\frac{V}{\alpha})^s$ of sets of size s, such that every set $Q \subseteq \mathcal{U}$ of size α has a subset in $\mathcal{T}$ (as guaranteed by Claim 2.4). If $\sum_{i=1}^{V} B_j \geq \alpha$, then there exists a set L of size α such that $B_j = 1$ for all $j \in L$. This implies that there exists a set $\tau \in \mathcal{T}$ such that $B_i = 1$ for all $i \in \tau$. By the union bound, the probability that the latter happens for some subset in $\mathcal{T}$ is at most

$$|\mathcal{T}|\Big(\frac{1}{V}\Big)^s \leq \alpha \log(V)\Big(\frac{V}{\alpha}\Big)^s\Big(\frac{1}{V}\Big)^s = \frac{\log(V)}{\alpha^{s-1}} \ .$$

□

2.3 Encryption Schemes and KDI Security

We define a (private-key) encryption scheme as a pair of an encryption and a decryption algorithm $(\mathrm{Enc}, \mathrm{Dec})$. On security parameter n, the encryption algorithm Enc gets as input a key of length $t(n)$ and a message of length m, and outputs a ciphertext of length $\ell(n, m)$. The decryption algorithm Dec, gets a key and a ciphertext and outputs the message.[4] Informally, an encryption scheme $(\mathrm{Enc}, \mathrm{Dec})$ is KDI-secure against a family of functions $\mathcal{H} \subseteq \{h : \{0,1\}^t \to \{0,1\}^*\}$, if no efficient adversary can distinguish between an oracle that correctly returns an encryption of $h(k)$, given input h, and one that returns an encryption of the all zero string of the same length. Note that if $\mathcal{H}$ contains functions that map to constants, plain-text queries can be obtained as well.

[4] In some definitions, a private-key encryption scheme also includes a key-generation algorithm "Gen". We omit this since we are not concerned by polynomial factors, and in this case one can simply take the random-coins used by Gen as the private key.

Definition 2.5 (KDI-security)**.** *Given an encryption scheme* $(\mathrm{Enc}, \mathrm{Dec})$ *and a key of length* t*, let* $Q^{\mathrm{Enc},k}$ *[resp.* $\widetilde{Q}^{\mathrm{Enc},k}$*] be an algorithm that gets as input a function* $h : \{0,1\}^t \mapsto \{0,1\}^{m(t)}$*, and returns* $\mathrm{Enc}(k, h(k))$ *[resp.* $\mathrm{Enc}(k, 0^{m(t)})$*] (if the schemes is randomized, it returns a random encryption). We say that* $(\mathrm{Enc}, \mathrm{Dec})$ *is* KDI-secure *for a class of functions* $\mathcal{H}$*, if*

$$\left|\mathsf{Pr}_{k\leftarrow\{0,1\}^{t(n)}}[\mathsf{A}^{Q^{\mathrm{Enc},k}}(1^n) = 1] - \mathsf{Pr}_{k\leftarrow\{0,1\}^{t(n)}}[\mathsf{A}^{\widetilde{Q}^{\mathrm{Enc},k}}(1^n) = 1]\right|$$

is negligible for every efficient algorithm A *that only queries functions in* $\mathcal{H}$*.*

2.4 Cryptographic Games

For reductions that treat the family $\mathcal{H}$ of query-functions as black-box, we are able to prove a very strong impossibility result. In this case, we show that essentially no cryptographic assumption is sufficient to guarantee the KDI-security of the scheme. In order to do this, we first define the set of cryptographic assumptions we consider.[5] For the sake of readability, however, we will not try to be as general as possible here. Yet, as far as we can see our definition still captures all natural hardness assumptions.

Definition 2.6 (cryptographic games)**.** *A* cryptographic game *is a (possibly inefficient) random system* Γ *that on security parameter* n *interacts with an attacker* A *and may output a special symbol* win*. In case* $\Gamma(1^n)$ *outputs this symbol in an interaction with* $\mathsf{A}(1^n)$*, we say that* $\mathsf{A}(\mathbf{1^n}) \leftrightarrow \mathbf{\Gamma}(\mathbf{1^n})$ wins*. The game is* secure *if* $\mathsf{Pr}[\mathsf{A}(1^n) \leftrightarrow \Gamma(1^n)$ *wins*$]$ *is negligible for all* PPT A*, where the probability is over the randomness of* A *and* Γ*.*

Examples: One might define the security of a one-way function f by the following game. On security parameter n, the system Γ selects a random $x \in \{0,1\}^n$ and sends $y = f(x)$ to the adversary. Γ outputs win if A outputs $x' \in f^{-1}(y)$.

To define the DDH hardness assumption one needs a bit more work.[6] On security parameter n, the system Γ expects first a sequence of at least n ones, we denote the actual number received by α. The system Γ then sends A a description of an appropriately chosen group $\langle g\rangle$ of order $\Omega(2^n)$ and the generator g, as well as α randomly chosen triples $(g^{x_i}, g^{y_i}, g^{z_i})$, where $z_i = x_i y_i$ or a uniform random element, each with probability $\frac{1}{2}$. The attacker A wins, if the number of instances where he incorrectly predicts whether z_i was chosen independently of x_i and y_i, is at most $\frac{\alpha}{2} - \alpha^{2/3}$.

Clearly, if Γ is secure, the usual DDH hardness assumption holds, since any algorithm that breaks the DDH assumption can be used for violating the security of Γ.[7] The reverse direction is implied by the main result (i.e., Theorem 1) of [IJK07] as follows. Assume the existence of a polynomial-time adversary A violating the security of Γ. Let $\varepsilon(n) \in 1/\operatorname{poly}(n)$ be such that A's winning probability is more than $\varepsilon(n)$ infinitely often and $\alpha < \frac{1}{\varepsilon(n)}$. Let $\delta = \frac{1}{2} - \varepsilon(n)$, $\gamma = \frac{1}{\alpha^{1/3}}$ and $k = \alpha$. Under the DDH assumption, [IJK07, Theorem 1] yields that the probability of A to be wrong on fewer than $T = (1-\gamma)\delta k = (1-\frac{1}{\alpha^{1/3}})(\frac{1}{2}-\varepsilon(n))\alpha > \frac{\alpha}{2} - \frac{\alpha}{2\alpha^{1/3}} - \alpha\varepsilon(n) > \frac{\alpha}{2} - \alpha^{2/3}$ of the games is bounded by $\varepsilon(n) > e^{-\Omega(\gamma^2\delta^2 k)}$, contradicting the existence of A.

2.5 Black-Box Reductions

A reduction from a primitive P to a primitive Q consists of showing that if there exists an implementation C of Q, then there exists an implementation M_C of P. This is equivalent to showing that for every adversary that breaks M_C, there exists an adversary that breaks C. Such a reduction is semi-black-box if it ignores the internal structure of Q's implementation, and it is fully-black-box (using the terminology of [RTV04]) if

[5] We remark that definitions of similar spirit to the one below were previously used in [DOP05, Hof08].

[6] The same argument can be applied for many other assumptions, but we refrain from formalizing this in order no to get bogged down in unrelated details.

[7] Given an algorithm A that breaks that DDH assumption with advantage $1/p(n)$ for infinitely many n, we construct the following attacker A′ for Γ. As first message A′ sends $1^{p(n)^3}$ to Γ, and then applies A for each of the triples it gets back from Γ.

it also has black-box proof of security. That is, the adversary for breaking Q ignores the internal structure of both Q's implementation and of the (alleged) adversary breaking P. The following definition expands the above general discussion for the case of a fully-black-box reduction of a KDI-secure encryption scheme from a one-way permutation.

Definition 2.7 (fully-black-box reduction)**.** *A fully-black-box reduction of a KDI-secure encryption scheme from a one-way permutation consists of polynomial-time oracle-aided algorithms* $(\mathrm{Enc}^{(\cdot)}, \mathrm{Dec}^{(\cdot)})$ *and a polynomial-time oracle-aided adversary* $\mathsf{A}_{\mathrm{OWP}}^{(\cdot)}$*, such that the following hold:*

- *If f is a permutation, then* $(\mathrm{Enc}^f, \mathrm{Dec}^f)$ *is an encryption scheme.*
- *For any (possibly unbounded)* $\mathsf{A}_{\mathrm{KDI}}$ *that breaks the KDI-security of the encryption scheme,* $\mathsf{A}_{\mathrm{OWP}}^{f,\mathsf{A}_{\mathrm{KDI}}}$ *inverts the permutation with non-negligible probability.*

When considering reductions from a KDI-secure cryptosystem, it is natural to consider whether the proof of security accesses the challenge functions also as a black box. We say that a proof of KDI-security of a cryptosystem is ***strongly-black-box***, if it treats the challenge function also as a black-box.

Definition 2.8 (strongly-black-box reduction)**.** *A reduction from a KDI-secure encryption scheme to a cryptographic game* Γ *with strongly-black-box proof of security, consists of polynomial-time oracle-aided algorithms* $(\mathrm{Enc}, \mathrm{Dec})$ *and a polynomial-time oracle-aided adversary* $\mathsf{A}_{\Gamma}^{(\cdot)}$ *such that the following holds:*

- $(\mathrm{Enc}, \mathrm{Dec})$ *is an encryption scheme.*
- *For any adversary* $\mathsf{A}_{\mathrm{KDI}}^{Q}$ *that breaks the KDI-security of* $(\mathrm{Enc}, \mathrm{Dec})$*, the oracle-aided adversary* $\mathsf{A}_{\Gamma}^{(\mathsf{A}_{\mathrm{KDI}})}$ *violates the security of* Γ*. Additionally,* A_{Γ} *treats the challenge functions provided by* $\mathsf{A}_{\mathrm{KDI}}$ *as a black box.*

The requirement that A_{Γ} *treats the challenge function as black-box, means that* A_{Γ} *can only obtain evaluations of it at arbitrary chosen points and the reduction must work for* every *challenge function (not just efficiently computable ones). In addition,* A_{Γ} *does not provide* Γ *with a description of the function.*[8]

2.6 Extending KDI-secure Encryption Schemes

We would like to make sure our impossibility results hold even for encryption schemes that encrypt messages of length one bit. For technical reasons, however, we will actually need to encrypt messages of length $2t$, where t is the key length. We therefore give a straightforward, but slightly tedious transformation that allows us to do that. (In fact, the following transformation does slightly more, in order to make the technical part in Sections 3 and 4 a bit easier.)

Proposition 2.9. *Let* $(\mathrm{Enc}, \mathrm{Dec})$ *be an encryption scheme for single bit messages. Assume* $(\mathrm{Enc}, \mathrm{Dec})$ *is KDI-secure for a given set* $\mathcal{H} \subseteq \{\{0,1\}^t \to \{0,1\}\}$*, then there exists an encryption scheme* $(\mathrm{Enc}_1, \mathrm{Dec}_1)$ *with the following properties:*

(a) *The key length* t_1 *of* $(\mathrm{Enc}_1, \mathrm{Dec}_1)$ *equals the security parameter.*

(b) $(\mathrm{Enc}_1, \mathrm{Dec}_1)$ *is defined for messages of arbitrary length.*

(c) $(\mathrm{Enc}_1, \mathrm{Dec}_1)$ *is KDI-secure for* $\mathcal{H}_1 := \left\{h : \{0,1\}^t \to \{0,1\}^* : \forall i, \forall \tau \in \{0,1\}^{t_1 - t} : h_{|i}(x,\tau) \in \mathcal{H}\right\}$*, where* $h_{|i}$ *is the function that outputs the i'th bit of the output of h.*[9]

(d) $(\mathrm{Enc}_1, \mathrm{Dec}_1)$ *has perfect correctness, i.e., it is always the case that* $(\mathrm{Dec}_1(k, \mathrm{Enc}_1(k, M)) = M$*.*

[8] Alternatively, given A_{Γ}'s (partial) view, it is possible to (efficiently) list all the queries done to the challenge function during the execution.

[9] Namely, $\mathcal{H}_1$ is the set of functions with the property that *every output bit* is described by a function in $\mathcal{H}$, after some appropriate padding of the input.

(e) $(\mathrm{Enc}_1, \mathrm{Dec}_1)$ *has deterministic decryption, i.e., the decryption procedure* Dec_1 *does not use any randomness.*

(f) *If* $(\mathrm{Enc}, \mathrm{Dec})$ *has a strongly-black-box [resp. black-box] proof of security to a cryptographic game* Γ*, then* $(\mathrm{Enc}_1, \mathrm{Dec}_1)$ *has a strongly-black-box [resp. black-box] proof of security to* Γ*.*

Remark 2.10. *Condition* (c) *gives a set* $\mathcal{H}_1$*, but one can always make this set smaller: if an encryption scheme is KDI-secure for* $\mathcal{H}$*, it is KDI-secure for* $\mathcal{H}^* \subseteq \mathcal{H}$*. Furthermore, if* $\mathcal{H}$ *is the set of all functions,* (c) *yields that* $\mathcal{H}_1$ *is the set of all functions, and if* $\mathcal{H}$ *is a set of* s*-wise independent functions,* (c) *can be used to get an efficient set of* s*-wise independent functions.*

We will assume that the security parameter σ of the given encryption scheme is *at least* the key length t. It is clear that $t \leq p(\sigma)$ for some polynomial p, and if $\sigma < t$, we simply redefine σ to be equal to t without changing the encryption scheme.

Proof. We assume without lost of generality that the security parameter σ of the given encryption scheme is *at least* the key length and start by building an encryption scheme $(\mathrm{Enc}_0, \mathrm{Dec}_0)$ that satisfies (d) and (e). For this, $(\mathrm{Enc}_0, \mathrm{Dec}_0)$ uses $(\mathrm{Enc}, \mathrm{Dec})$ (with the same security parameter) as a subroutine and works as follows: on inputs (k, m), algorithm Enc_0 chooses, in addition to r_{Enc}- the random-coins needed by Enc, also the random-coins to be used by Dec, r_{Dec}. It then checks whether $\mathrm{Dec}(r_{\mathrm{Dec}}, k, \mathrm{Enc}(r_{\mathrm{Enc}}, k, m)) = m$. If this holds, then $\mathrm{Enc}_0(k, m)$ outputs $0 \circ \mathrm{Enc}(k, m) \circ r_{\mathrm{Dec}}$, otherwise it outputs $1 \circ m$ (where $\circ$ denotes the concatenation of strings). On ciphertext $c' = (0 \circ c \circ r_{\mathrm{Dec}})$, algorithm $\mathrm{Dec}_0(k, c')$ outputs $\mathrm{Dec}(r_{\mathrm{Dec}}, k, c)$, where on ciphertext $c' = (1 \circ m)$ it outputs m.

Having the above, we define $(\mathrm{Enc}_1, \mathrm{Dec}_1)$ as follows: given a message to encrypt of arbitrary length, we partition the message into bits and encrypt every bit using Enc_0 (for the same security parameter). In the decryption we do the same using Dec_0. Additionally, if the security parameter σ of $(\mathrm{Enc}, \mathrm{Dec})$ is greater than the key length t, algorithm Enc_1 uses $\sigma - t$ additional key bits that it ignores. All in all, $(\mathrm{Enc}_1, \mathrm{Dec}_1)$ has a deterministic encryption algorithm and is always correct. Clearly, $(\mathrm{Enc}_1, \mathrm{Dec}_1)$ satisfies conditions (a) and (b).

We now show that condition (c) is also satisfied by $(\mathrm{Enc}_1, \mathrm{Dec}_1)$: we construct an adversary $\mathsf{A}^{(\cdot)}$ for $(\mathrm{Enc}, \mathrm{Dec})$ given an adversary for $(\mathrm{Enc}_1, \mathrm{Dec}_1)$ as follows. Let A_1 be an adversary for $(\mathrm{Enc}_1, \mathrm{Dec}_1)$. The adversary $\mathsf{A}^{\mathsf{A}_1}$ first picks $t_1 - t$ additional key bits, and then emulates A_1. Every time A_1 asks for an encryption of a function $h \in \mathcal{H}^*$, $\mathsf{A}^{\mathsf{A}_1}$ decomposes h into its components $h_i \in \mathcal{H}$. If $t_1 > t$ it computes the respective functions in $\mathcal{H}$, and then obtains each of the encryptions from the given KDI-encryption oracle of $(\mathrm{Enc}, \mathrm{Dec})$. Note that this decomposition can be done in a black-box manner, by considering for each h_i only the relevant part in the output of h. Then, $\mathsf{A}^{\mathsf{A}_1}$ prefixes each answer with a 0 (to denote that the decryption is correct), adds uniformly chosen randomness for the decryption, concatenates the answers and returns the resulting ciphertext to A_1. One checks that with high probability the distribution of the resulting ciphertext is as required. Once A_1 gives a prediction, $\mathsf{A}^{\mathsf{A}_1}$ outputs this prediction.

Finally, we consider condition (f). Let $\mathsf{A}_\Gamma^{(\cdot)}$ be the adversary, guaranteed by the black-box proof of security of $(\mathrm{Enc}, \mathrm{Dec})$, that violates the security of Γ given an adversary for $(\mathrm{Enc}, \mathrm{Dec})$. It is clear that $\mathsf{A}_\Gamma^{\mathsf{A}^{(\cdot)}}$ is a black-box proof of security for $(\mathrm{Enc}_1, \mathrm{Dec}_1)$. Moreover, assuming that $(\mathrm{Enc}, \mathrm{Dec})$ has a strongly-black-box proof of security (and that $\mathsf{A}_\Gamma^{(\cdot)}$ is the adversary guaranteed by this proof), it is easy to verify that $\mathsf{A}_\Gamma^{\mathsf{A}^{(\cdot)}}$ is a strongly-black-box proof of security for $(\mathrm{Enc}_1, \mathrm{Dec}_1)$. □

3 From One-way Permutations

In this section we prove Theorem 1.1, but we only give the proof for the case of one-way permutations. The proof for (enhanced) family of trapdoor permutations follows immediately using standard techniques (cf., [GT00, HHRS07]). Let $(\mathrm{Enc}^{(\cdot)}, \mathrm{Dec}^{(\cdot)})$ be an encryption scheme with oracle access to a one-way permutation. By Proposition 2.9, we can assume that the encryption scheme is always correct, has a deterministic

decryption algorithm, defined on messages of any polynomial length and has a security parameter t equal to it's key length. We let $\ell(t)$ be the length of an encryption of a message of length $2t$. In order to prove Theorem 1.1, we use the following inefficient algorithm $\text{Breaker}^{f,h}$.

Algorithm 3.1. $\text{Breaker}^{f,h}$.

Oracles: *A function* $f : \{0,1\}^t \times \{0,1\}^{\ell(t)} \mapsto \{0,1\}^{2t}$ *(defined for every* $t \in \mathbb{N}$*) and an infinite sequence of functions* $h = \left\{h_t : \{0,1\}^t \mapsto \{0,1\}^{2t}\right\}_{t\in\mathbb{N}}$.

Input: *A pair* $(t, c) \in \mathbb{N} \times \{0,1\}^*$.

Operation: *Return the smallest* $k \in \{0,1\}^t$ *such that* $f(k, c) = h_t(k)$*, or* $\perp$ *if no such* k *exists.*

Let $\Pi = \{\Pi_t\}_{t\in\mathbb{N}}$, where Π_t is the set of all possible permutations over $\{0,1\}^t$, and let $\mathcal{H} = \{\mathcal{H}_t\}_{t\in\mathbb{N}}$, where h_t is a family of $(\ell(t) + t)$-wise independent hash functions from $\{0,1\}^t$ to $\{0,1\}^{2t}$ with polynomial description size. We denote by $\pi = \{\pi_t\}_{t\in\mathbb{N}} \leftarrow \Pi$ [resp., $h = \{h_t\}_{t\in\mathbb{N}} \leftarrow \mathcal{H}$] the sequence of functions induced by selecting, for every $t \in \mathbb{N}$, π_t uniformly at random from Π_t [resp., h_t uniformly at random from $\mathcal{H}_t$]. In this section, we consider an instantiation of Breaker with $f = \text{Dec}^\pi$, where π is chosen at random from Π, and h chosen at random from $\mathcal{H}$. In Section 3.1, we show how to use $\text{Breaker}^{\text{Dec}^\pi,h}$ for violating the KDI-security of $(\text{Enc}^\pi, \text{Dec}^\pi)$, where in Section 3.2 we show that $\text{Breaker}^{\text{Dec}^\pi,h}$ does not help inverting a random π. We combine these two facts in Section 3.3 for proving Theorem 1.1.

3.1 Breaker Violates the KDI-Security of the Scheme

The following adversary uses $\text{Breaker}^{\text{Dec}^\pi,h}$ for breaking the KDI-security of $(\text{Enc}^\pi, \text{Dec}^\pi)$.

Algorithm 3.2. *Algorithm* $\mathsf{A}_{\text{KDI}}^{\text{Breaker}^{\text{Dec}^\pi,h},h}$.

Oracles: *An infinite sequence of functions* $h = \left\{h_t : \{0,1\}^t \to \{0,1\}^{2t}\right\}_{t\in\mathbb{N}}$ *and* $\text{Breaker}^{\text{Dec}^\pi,h}$.

Input: *Security parameter* t.

Operation:

Step 1: *Call* $Q(h_t)$ *(or* $\widetilde{Q}(h_t)$*) to obtain an encryption* c *of* $h_t(k)$ *(or of* 0^{2t}*).*

Step 2: *Call* $\text{Breaker}^{\text{Dec}^\pi,h}(t, c)$ *to obtain a candidate key* k' *or* $\perp$.

Step 3: *Output* 1 *iff* Breaker *did not return* $\perp$.

Lemma 3.3. *For every value of* $\pi \in \Pi$*, algorithm* $\mathsf{A}_{\text{KDI}}^{\text{Breaker}^{\text{Dec}^\pi,h},h}$ *breaks the KDI-security of the* $(\text{Enc}^\pi, \text{Dec}^\pi)$ *with probability one over a random choice of* $h \in \mathcal{H}$.

Proof. Algorithm A_{KDI} only gives the wrong answer if the oracle is $\widetilde{Q}$ and Breaker does not return $\perp$. Assume now that the oracle is $\widetilde{Q}$. Then, for any fixed k' and k we have $\mathsf{Pr}_{h_t}[h_t(k') = \text{Dec}(k', \text{Enc}(k, 0^{2t}))] = \frac{1}{2^{2t}}$, and using the union bound we have that the for a fixed k the probability that Breaker does not return $\perp$ is at most 2^{-t}.[10] Using an averaging argument, the probability that h_t is such that something else but $\perp$ is returned with probability higher than $2^{-t/2}$ is at most $2^{-t/2}$.

Since the $h_t \in h$'s are chosen independently from each other, the probability that there exists $t_0 \in \mathbb{N}$ for which A_{KDI} breaks the scheme for no $t > t_0$ is zero. We conclude that with probability one over the random choice of $h \in \mathcal{H}$, it holds that A_{KDI} breaks the KDI-security of (Enc, Dec) infinitely often. □

[10] For this lemma, we are only using the "one-wise" independence of h.

3.2 Breaker Does not Invert Random Permutations

We prove the following upper bound on the probability that an algorithm with access to Breaker inverts a random permutation. In the following let $\mu_{\mathsf{A}}(n)$ be an upper bound on number of π queries and the length of the maximal π query that A does on input $y \in \{0,1\}^n$ (either directly or through the calls to $\text{Breaker}^{\text{Dec}^\pi,h}$), and let $\mu_{\text{Dec}}(t)$ the same bound with respect to the π queries that Dec does on input $(k,c) \in \{0,1\}^t \times \{0,1\}^{\ell(t)}$. We assume without loss of generality that both upper bounds are monotonically increasing, that $\mu_{\text{Dec}}(t) \geq t + \ell(t)$ and that $\mu_{\mathsf{A}}(n) \geq n$. We also assume that $\mu_{\text{Dec}}(t) < 2^t$.

Lemma 3.4. *Let A be an adversary that gets h as an auxiliary input*[11] *and has oracle access to π and $\text{Breaker}^{\text{Dec}^\pi,h}$. Then for every $y \in \{0,1\}^n$ it holds that*

$$\mathsf{Pr}_{\pi \leftarrow \Pi, h \leftarrow \mathcal{H}}[\mathsf{A}_h^{(\pi,\text{Breaker}^{\text{Dec}^\pi,h})}(y) = \pi^{-1}(y)] < 3\mu_{\mathsf{A}}(n)\left(2^{-\mu_{\text{Dec}}^{-1}(n)} + \mu_{\text{Dec}}(\mu_{\mathsf{A}}(n))^2 2^{-n}\right) \ ,$$

where $\mu_{\text{Dec}}^{-1}(n) := \min\{t \in \mathbb{N} \ : \ \mu_{\text{Dec}}(t) \geq n\}$.

Applying the Borel-Cantelli lemma on the above we get the following corollary.

Corollary 3.5. *Assume that A and* Dec *are polynomially bounded, then there exists a negligible function ε such that with probability one over the choice of π and h, $\mathsf{Pr}_{y \leftarrow \{0,1\}^n}[\mathsf{A}_h^{(\pi,\text{Breaker}^{\text{Dec}^\pi,h})}(y) = \pi^{-1}(y)] < \varepsilon(n)$ for large enough n.*

In Appendix A, we give a proof of a non-uniform version of Lemma 3.4 (the adversary can use an arbitrary additional non-uniform advice) using the technique introduced by Gennaro and Trevisan [GT00]. Here, we use a different technique that is similar to the one used by Simon [Sim98]. The main idea is to study what happens if π is modified slightly by mapping a second, randomly chosen element to y (the element that A tries to invert). We show that such a change will likely go unnoticed by $\mathsf{A}(y)$, and it will not find the new preimage. After the change, however, both preimages of y are equally likely to be the original one, so $\mathsf{A}(y)$ could not have found the original one either.[12]

For a given function $g : \{0,1\}^n \mapsto \{0,1\}^n$ and two strings $x^*, y \in \{0,1\}^n$, we define the function $g_{|x^* \to y}$ as

$$g_{|x^* \to y}(x) := \begin{cases} y & \text{if } x = x^*, \\ g(x) & \text{otherwise.} \end{cases}$$

We assume that all calls to $\text{Dec}^{\pi_{|x^* \to y}}$ are well defined. In particular, if Dec queries $\pi_{|x^* \to y}$ both at position $\pi^{-1}(y)$ and at $x^* \neq \pi^{-1}(y)$ (and thus might act arbitrarily as it "notices" that $\pi_{|x^* \to y}$ is not a permutation), we assume it stops and outputs 0. We now wish to consider the elements $\{x^* \in \{0,1\}^n\}$ for which $\text{Breaker}^{\text{Dec}^\pi,h}(t,c) \neq \text{Breaker}^{\pi_{|x^* \to y},h}(t,c)$. The set $\text{Diff}^\pi(t,c,h,y)$ is a (possibly proper) superset of this set.

Definition 3.6 (Diff)**.** *For an oracle function* Dec*, an infinite sequence of functions $h \in \mathcal{H}$, $t \in \mathbb{N}$, and strings $c \in \{0,1\}^*$ and $y \in \{0,1\}^n$, we let*

$$\text{Diff}^\pi(t,c,h,y) := \Big\{ x^* \in \{0,1\}^n \ \Big| \ \exists k \in \{0,1\}^t \ : \ \big(\text{Dec}^\pi(k,c) \neq h_t(k) = \text{Dec}^{\pi_{|x^* \to y}}(k,c)\big) \\ \vee \big(\text{Dec}^\pi(k,c) = h_t(k) \neq \text{Dec}^{\pi_{|x^* \to y}}(k,c)\big) \Big\}.$$

[11] We handle the fact that h is an infinite object, by only providing A the (description of the) first $q(n)$ functions in the sequence, where $q(n)$ is an upper bound on the running-time of $\mathsf{A}(y \in \{0,1\}^n)$.

[12] The main difference between our approach and the one in [Sim98], is that we do not insist on keeping π a permutation. It turns out that this slackness makes our proof significantly simpler.

Note that for $x^* \notin \mathrm{Diff}^{\pi}(t,c,h,y)$, it holds that $\mathrm{Breaker}^{\mathrm{Dec}^{\pi},h}(t,c) = \mathrm{Breaker}^{\pi_{|x^*\to y},h}(t,c)$. To see this, let $k_0 \neq \perp$ be the lexicographic smaller output of the two calls. Clearly, k_0 must be the output of both calls to Breaker. The next claim states that if h is uniformly chosen from $\mathcal{H}$, then $\mathrm{Diff}^{\pi}(t,c,h,y)$ is very likely to be small for all possible c.

Claim 3.7. *Let* A *be a deterministic adversary with oracle access to* π *and* $\mathrm{Breaker}^{\mathrm{Dec}^{\pi},h}$*, which gets* h *as an auxiliary input. The following holds for every* π *and* $y \in \{0,1\}^n$*:*

$$\mathsf{Pr}_{h\leftarrow\mathcal{H}}\left[\mathsf{A}_h^{(\pi,\mathrm{Breaker}^{\mathrm{Dec}^{\pi},h})}(y) \text{ queries } \mathrm{Breaker}^{\mathrm{Dec}^{\pi},h}(t,c) \text{ with } |\mathrm{Diff}^{\pi}(t,c,h,y)| \geq \mu_{\mathrm{Dec}}(\mu_{\mathsf{A}}(n))^2\right] < \mu_{\mathsf{A}}(n)2^{-\mu_{\mathrm{Dec}}^{-1}(n)}$$

Proof. For $t \in \mathbb{N}$, $c \in \{0,1\}^*$ and $k \in \{0,1\}^t$, let $\mathcal{D}_{c,k}$ be the set of all possible images of $\mathrm{Dec}^{\pi_{|x^*\to y}}(k,c)$, enumerating over all $x^* \in \{0,1\}^n$ (i.e., the set $\mathcal{D}_{c,k} := \{\mathrm{Dec}^{\pi_{|x^*\to y}}(k,c) \,:\, x^* \in \{0,1\}^n\}$).[13] We first note that $|\mathcal{D}_{c,k}| \leq \mu_{\mathrm{Dec}}(t)+1 \leq 2^t$ – in an execution of $\mathrm{Dec}^{\pi}(k,c)$ at most $\mu_{\mathrm{Dec}}(t)$ elements $x_1,\ldots,x_{\mu_{\mathrm{Dec}}(t)}$ are queried, and only if $x^* \in \{x_1,\ldots,x_{\mu_{\mathrm{Dec}}(t)}\}$ the image of Dec can be changed. Let $\mathcal{H}_t$ be the t'th entry in $\mathcal{H}$. Applying Lemma 2.3 with $V = 2^t$, $s = t+\ell(t)$, $\alpha = \mu_{\mathrm{Dec}}(t)$ and letting $B_k = 1$ iff $h_t(k) \in \mathcal{D}_{c,k}$, we have that

$$\mathsf{Pr}_{h_t\leftarrow\mathcal{H}_t}\left[\left|\{k \in \{0,1\}^t \,:\, h_t(k) \in \mathcal{D}_{c,k}\}\right| \geq \mu_{\mathrm{Dec}}(t)\right] < \frac{t}{\mu_{\mathrm{Dec}}(t)^{t+\ell(t)-1}} \leq 2^{-t-\ell(t)} \tag{1}$$

We next show that

$$|\,\mathrm{Diff}^{\pi}(t,c,h,y)| \leq \mu_{\mathrm{Dec}}(t)\cdot\left|\left\{k \in \{0,1\}^t \,:\, h_t(k) \in \mathcal{D}_{c,k}\right\}\right| \tag{2}$$

We prove Equation (2) by presenting an injective function ϕ from $\mathrm{Diff}^{\pi}(t,c,h,y)$ to $\{1,\ldots,\mu_{\mathrm{Dec}}(t)\} \times \{k : h_t(k) \in \mathcal{D}_{c,k}\}$. If $x^* \in \mathrm{Diff}^{\pi}(t,c,h,y)$, then there exists $k_{x^*} \in \{0,1\}^t$ such that $\mathrm{Dec}^{\pi}(k_{x^*},c) \neq h_t(k_{x^*}) = \mathrm{Dec}^{\pi_{|x^*\to y}}(k_{x^*},c)$ or $\mathrm{Dec}^{\pi}(k_{x^*},c) = h_t(k_{x^*}) \neq \mathrm{Dec}^{\pi_{|x^*\to y}}(k_{x^*},c)$ and therefore $k_{x^*} \in \{k : h_t(k) \in \mathcal{D}_{c,k}\}$. Furthermore, $\mathrm{Dec}^{\pi}(k_{x^*},c)$ must query π on x^*. Let i_{x^*} denote the index (i.e., position) of the query $\pi(x^*)$ among the π queries that $\mathrm{Dec}^{\pi}(k_{x^*},c)$ does, and let ϕ be the function that maps x^* to (i_{x^*},k_{x^*}). Since a pair (i,k) specifies a single x (the one queried at i'th position in $\mathrm{Dec}^{\pi}(k,c)$), it follows that ϕ is indeed injective.

We note that for $t > \mu_{\mathrm{Dec}}^{-1}(n)$, it always holds that $x^* \notin \mathrm{Diff}^{\pi}(t,c,h,y)$ (Dec cannot invoke π on such a long input). Combining (1) and (2), we have that with probability at least $1-\mu_{\mathsf{A}}(n)2^{-\mu_{\mathrm{Dec}}^{-1}(n)}$ over the choice of h, for each of the at most $\mu_{\mathsf{A}}(n)$ queries $\mathrm{Breaker}(t,c)$ that $\mathsf{A}(y)$ does, it holds that $|\mathrm{Diff}(t,c,h,y)| \leq \mu_{\mathrm{Dec}}(t)^2 \leq \mu_{\mathrm{Dec}}(\mu_{\mathsf{A}}(n))^2$. □

For a given value of π and $x_0,x_1 \in \{0,1\}^n$, let $\pi_{|x_0\leftrightarrow x_1} := \pi_{|x_0\to\pi(x_1),x_1\to\pi(x_0)}$. We next show that with high probability, $\mathsf{A}(y)$ behaves exactly the same given the oracle π or $\pi_{|x^*\leftrightarrow y}$.

Definition 3.8 (trace)**.** *For a given oracle function* Dec*, the* trace*,* $\mathrm{tr}(\pi,h,y,r_{\mathsf{A}})$*, of an adversary* A *is the sequence of all queries* $\mathsf{A}(y)$ *makes to* $\mathrm{Breaker}^{\mathrm{Dec}^{\pi},h}$ *and* π *(and their responses), when it uses* r_{A} *as its random-coins and gets* h *as an auxiliary input.*

Claim 3.9. *Let* $y \in \{0,1\}^n$ *and let* A *be an adversary with oracle access to* π *and* $\mathrm{Breaker}^{\mathrm{Dec}^{\pi},h}$*, and assume that* A *queries* π *on its output before returning it. Then, the following holds*

$$\mathsf{Pr}_{\pi,h,r_{\mathsf{A}},x^*\leftarrow\{0,1\}^n}[\mathrm{tr}(\pi,h,y,r_{\mathsf{A}}) \neq \mathrm{tr}(\pi_{|x^*\leftrightarrow\pi^{-1}(y)},h,y,r_{\mathsf{A}})] < p(n) \ ,$$

where $p(n) := 2\mu_{\mathsf{A}}(n)\left(2^{-\mu_{\mathrm{Dec}}^{-1}(n)} + \mu_{\mathrm{Dec}}(\mu_{\mathsf{A}}(n))^2 2^{-n}\right)$.

Proof. Let $\mathcal{X}_{\pi}$ be all the queries to π in $\mathrm{tr}(\pi,h,y,r_{\mathsf{A}})$, clearly $|\mathcal{X}_{\pi}| \leq \mu_{\mathsf{A}}(n)$. Let further $\mathcal{X}_{\mathrm{Diff}}$ be the union of the sets $\mathrm{Diff}^{\pi}(t,c,h,y)$ for all calls (t,c) made by A to Breaker. If $x^* \notin \mathcal{X}_{\pi} \cup \mathcal{X}_{\mathrm{Diff}}$, we have that $\mathrm{tr}(\pi,h,y,r_{\mathsf{A}}) = \mathrm{tr}(\pi_{|x^*\to y},h,y,r_{\mathsf{A}})$ (cf., the remark after Definition 3.6). Claim 3.7 yields that with probability at least

[13] Note that the original image $\mathrm{Dec}^{\pi}(k,c)$ is in $\mathcal{D}_{c,k}$.

$1 - \mu_{\mathsf{A}}(n)2^{-\mu_{\mathrm{Dec}}^{-1}(n)}$ over the choice of h and r_A, it holds that $|\mathcal{X}_\pi \cup \mathcal{X}_{\mathrm{Diff}}| \leq \mu_{\mathsf{A}}(n)\mu_{\mathrm{Dec}}(\mu_{\mathsf{A}}(n))^2$. Hence, a union bound yields that

$$\mathsf{Pr}_{h,r_{\mathsf{A}},x^* \leftarrow \{0,1\}^n}[\mathrm{tr}(\pi,h,y,r_{\mathsf{A}}) \neq \mathrm{tr}(\pi_{|x^* \to y},h,y,r_{\mathsf{A}})] < \mu_{\mathsf{A}}(n)\left(2^{-\mu_{\mathrm{Dec}}^{-1}(n)} + \mu_{\mathrm{Dec}}(\mu_{\mathsf{A}}(n))^2 2^{-n}\right) \quad (3)$$

Finally, if $\mathrm{tr}(\pi,h,y,r_{\mathsf{A}})$ and $\mathrm{tr}(\pi_{|x^* \leftrightarrow \pi^{-1}(y)},h,y,r_{\mathsf{A}})$ are different, then one of $\mathrm{tr}(\pi,h,y,r_{\mathsf{A}}) \neq \mathrm{tr}(\pi_{|x^* \to y},h,y,r_{\mathsf{A}})$ or $\mathrm{tr}(\pi_{|x^* \to y},h,y,r_{\mathsf{A}}) \neq \mathrm{tr}(\pi_{|x^* \leftrightarrow \pi^{-1}(y)},h,y,r_{\mathsf{A}})$ must hold. Since $\pi_{|x \leftrightarrow \pi^{-1}(y)}$ is also a permutation, and x^* is a uniformly chosen element given $\pi_{|x^* \leftrightarrow \pi^{-1}(y)}$ and y, Equation (3) states that both these events have probability at most $p(n)/2$. □

We now use Claim 3.9 for proving Lemma 3.4.

Proof. (of Lemma 3.4) Assuming without lost of generality that A queries it's output, the probability that $\mathsf{A}^{(\pi,\mathrm{Breaker}^{\mathrm{Dec}^\pi,h})}(y)_h = \pi^{-1}(y)$ is at most the probability that the traces $\mathrm{tr}(\pi,h,y,r_{\mathsf{A}})$ and $\mathrm{tr}(\pi_{|x^* \leftrightarrow \pi^{-1}(y)},h,y,r_{\mathsf{A}})$ are different plus 2^{-n} (to handle the case $x^* = \pi^{-1}(y)$). By Claim 3.9, the latter probability is at most $2^{-n} + p(n)$. □

3.3 Putting it Together

We use the results from Section 3.1 and Section 3.2 for proving Theorem 1.1.

Proof. (of Theorem 1.1) Assume that there exists a black-box proof of security from breaking the KDI-security of $(\mathrm{Enc}^\pi, \mathrm{Dec}^\pi)$ using a poly-wise independent hash function to breaking the hardness of π, and let $M^{(\cdot)}$ be the algorithm for inverting π as guaranteed by this proof of security. Lemma 3.3 yields that $\mathsf{A}_{\mathrm{KDI}}^{\mathrm{Breaker}^{\mathrm{Dec}^\pi,h},h}$ breaks the KDI-security of $(\mathrm{Enc}^\pi, \mathrm{Dec}^\pi)$ with probability one over the choice of h. Thus, $M^{\mathsf{A}_{\mathrm{KDI}}^{\mathrm{Breaker}^{\mathrm{Dec}^\pi,h},h}}$ needs to break the one-way property of π with probability one over the choice of h as well. However, since $\mathsf{A}^{\pi,\mathsf{B}}$ can be efficiently emulated by an algorithm $\widetilde{\mathsf{A}}$ with oracle access to $\mathrm{Breaker}^{\mathrm{Dec}^\pi,h}$ and π, and given h as an auxiliary input, Corollary 3.5 yields that with probability one over the choice of π and h, algorithm $\mathsf{A}^{\pi,\mathsf{B}}$ does not break the one-wayness of π, and a contradiction is derived. □

4 From Arbitrary Assumptions

In this section, we rule out the existence of reductions with strongly-black-box proof of security from the KDI-security of an encryption scheme, to a very large class of hardness assumptions. That is, we prove the following theorem.

Theorem 4.1 (restatement of Theorem 1.2)**.** *There exists no reduction with strongly-black-box proof of security from the KDI-security of an encryption scheme to* any *cryptographic game.*

Let $(\mathrm{Enc}, \mathrm{Dec})$ be an encryption scheme. As in Section 3, we use Proposition 2.9 and assume without loss of generality that the encryption scheme is always correct, has deterministic decryption algorithm, is defined on messages of any length, and has security parameter t equal to the key length. We let $\ell(t)$ be the length of an encryption of a message of length $2t$.

Consider an instantiation of Breaker (Algorithm 3.1) with $f = \mathrm{Dec}$ and $h \in \mathcal{H} = \{\mathcal{H}_t\}_{t \in \mathbb{N}}$, where $\mathcal{H}_t$ is the set of *all* possible function from $\{0,1\}^t$ to $\{0,1\}^{2t}$. As in Section 3, we have that there exists an efficient algorithm, with oracle access to $\mathrm{Breaker}^{\mathrm{Dec},h}$ and h, that breaks the KDI-security of $(\mathrm{Enc}, \mathrm{Dec})$ with probability one over the choice of $h \in \mathcal{H}$. The following Lemma states that in many settings, having oracle access to $\mathrm{Breaker}^{\mathrm{Dec},h}$ does not yield any significant additional power.

Lemma 4.2. *Let* $\mathsf{A}^{\mathrm{Breaker}^{\mathrm{Dec},h},h}$ *be an algorithm with oracle access to* $\mathrm{Breaker}^{\mathrm{Dec},h}$ *and* h*, and let* $t_{\mathsf{A}}(n)$*, for security parameter* n*, be a polynomial-time computable upper bound on the running-time of* $\mathsf{A}^{\mathrm{Breaker}^{\mathrm{Dec},h},h}$*.*

Then for every polynomial computable function $\delta : \mathbb{N} \mapsto [0,1]$, there exists an algorithm $\widetilde{A}^h_\delta$, which has oracle access only to h, runs in time $\mathrm{poly}(1/\delta(n), t_{\mathsf{A}}(n), n)$ and uses random-coins of the same length as $\mathsf{A}^{\mathrm{Breaker}^{\mathrm{Dec},h},h}$ such that the following holds. If $\mathsf{A}^{\mathrm{Breaker}^{\mathrm{Dec},h},h}$ and $\widetilde{A}^h_\delta$ are using the same random-coins, $\mathsf{A}^{\mathrm{Breaker}^{\mathrm{Dec},h},h}(1^n) = \widetilde{A}^h_\delta(1^n)$ with probability $1-\delta(n)$ over a random choice of h.

Proof. Algorithm $\widetilde{A}$ emulates A, using its random-coins as the random-coins of A, while remembering all query and answer pairs to h. When the emulated A queries $\mathrm{Breaker}^{\mathrm{Dec},h}(t,c)$, algorithm $\widetilde{A}$ distinguishes between the following two cases:

Case 1: $t < \log(t_{\mathsf{A}}(n)) + \log(1/\delta(n))$**.** $\widetilde{A}$ fully emulates $\mathrm{Breaker}^{\mathrm{Dec},h}$. Namely, $\widetilde{A}$ evaluates $h_t(k)$ for all $k \in \{0,1\}^t$ and returns the first one for which $\mathrm{Dec}(k,c) = h_t(k)$. It returns $\perp$ if no such k exists.

Case 2: $t \geq \log(t_{\mathsf{A}}(n)) + \log(1/\delta(n))$**.** $\widetilde{A}$ checks all the previous queries to h of length t in lexicographic order. If for one of those queries it holds that $\mathrm{Dec}(k,c) = h_t(k)$, it returns k, otherwise it returns $\perp$.

The bound on the running-time of $\widetilde{A}$ is clear, in the following we show $\widetilde{A}$ emulates A well. We first note that in Case 1, $\widetilde{A}$ always returns the same answer that $\mathrm{Breaker}^{\mathrm{Dec},h}$ would. To handle Case 2, let $k \in \{0,1\}^t$ and assume that the query $h_t(k)$ was not perviously asked by A. Since h is length doubling, the probability over the choice of h that $\mathrm{Dec}(k,c) = h_t(k)$ is 2^{-2t}. Using a union bound we have that the probability, over the choice of h, that $\widetilde{A}$ returns a value different from what $\mathrm{Breaker}^{\mathrm{Dec},h}$ would (i.e., $\widetilde{A}$ returns $\perp$ where $\mathrm{Breaker}^{\mathrm{Dec},h}$ finds a consistent key) is at most $2^{-\log(t_{\mathsf{A}}(n))-\log(1/\delta(n))} = \delta(n)/t_{\mathsf{A}}(n)$. Since there are at most $t_{\mathsf{A}}(n)$ calls to $\mathrm{Breaker}^{\mathrm{Dec},h}$, the probability that in any of those $\widetilde{A}$ returns a wrong value is at most $\delta(n)$, which proves the lemma. □

We can now prove Theorem 4.1.

Proof. (of Theorem 4.1) The proof follows the lines of the one of Theorem 1.1, but we need to work a little harder for proving that having access to h does not give the adversary additional power.[14]

Assume that there exists a strongly-black-box proof of security from $(\mathrm{Enc},\mathrm{Dec})$ to a cryptographic game Γ and let $M^{(\cdot)}$ be the algorithm for breaking Γ as guaranteed by this proof of security. It easily follows from the proof of Lemma 3.3, that also in the setting of this section there exists an efficient algorithm $\mathsf{A}_{\mathrm{KDI}}^{\mathrm{Breaker}^{\mathrm{Dec},h},h}$, with oracle access to $\mathrm{Breaker}^{\mathrm{Dec},h}$ and h, that breaks the KDI-security of $(\mathrm{Enc},\mathrm{Dec})$ with probability one over the choice of h. It follows that $M^{\mathsf{A}_{\mathrm{KDI}}^{\mathrm{Breaker}^{\mathrm{Dec},h},h}}$ breaks the security of Γ with probability one over the choice of h. Namely,

$$\mathsf{Pr}_h\Big[\mathsf{Pr}_{r_{\mathsf{A}},r_\Gamma}[M^{\mathsf{A}_{\mathrm{KDI}}^{\mathrm{Breaker}^{\mathrm{Dec},h},h}} \leftrightarrow \Gamma(1^n) \text{ wins}] > \frac{1}{p_h(n)} \qquad \text{for infinitely many } n\Big] = 1 \ , \tag{4}$$

where r_{A} and r_Γ denote the random-coins of A and Γ, respectively, and p_h is some polynomial that may depend on h. In the following we first remove the dependence of the polynomial p_h from h. For this let

$$\varepsilon(n) := \mathsf{Pr}_{h,r_{\mathsf{A}},r_\Gamma}[M^{\mathsf{A}_{\mathrm{KDI}}^{\mathrm{Breaker}^{\mathrm{Dec},h},h}} \leftrightarrow \Gamma(1^n) \text{ wins}] = \mathsf{E}_h\Big[\mathsf{Pr}_{r_{\mathsf{A}},r_\Gamma}[M^{\mathsf{A}_{\mathrm{KDI}}^{\mathrm{Breaker}^{\mathrm{Dec},h},h}} \leftrightarrow \Gamma(1^n) \text{ wins}]\Big] \ .$$

We show that ε is non negligible. Using Markov's inequality we get for every $n \in \mathbb{N}$ that

$$\mathsf{Pr}_h\Big[\mathsf{Pr}_{r_{\mathsf{A}},r_\Gamma}[M^{\mathsf{A}_{\mathrm{KDI}}^{\mathrm{Breaker}^{\mathrm{Dec},h},h}} \leftrightarrow \Gamma(1^n) \text{ wins}] < n^2\varepsilon(n)\Big] > 1 - \frac{1}{n^2} \ ,$$

and therefore[15]

$$\mathsf{Pr}_h\Big[\mathsf{Pr}_{r_{\mathsf{A}},r_\Gamma}[M^{\mathsf{A}_{\mathrm{KDI}}^{\mathrm{Breaker}^{\mathrm{Dec},h},h}} \leftrightarrow \Gamma(1^n) \text{ wins}] < n^2\varepsilon(n) \qquad \text{for all } n > 2\Big] \geq 1 - \sum_{n=2}^{\infty}\frac{1}{n^2} > \frac{1}{3} \tag{5}$$

[14] One gets this property "for free", when the underlying hardness assumption is inverting a random permutation.

[15] The σ-additivity of the measure implies that the event in the next probability is measurable.

Combining Equations (4) and (5) we get that

$$\mathsf{Pr}_h\Big[\frac{1}{p_h(n)} < \mathsf{Pr}_{r_{\mathsf{A}},r_\Gamma}[M^{\mathsf{A}_{\mathrm{KDI}}^{\mathrm{Breaker}^{\mathrm{Dec},h},h}} \leftrightarrow \Gamma(1^n) \text{ wins}] < n^2\varepsilon(n) \qquad \text{for infinitely many } n\Big] > \frac{1}{3}\ , \tag{6}$$

which implies that there is a polynomial $p(n)$ such that $\varepsilon(n) > \frac{1}{p(n)}$ infinitely often.

In order to finish the proof, we would like now to apply Lemma 4.2 on $M^{\mathsf{A}_{\mathrm{KDI}}^{\mathrm{Breaker}^{\mathrm{Dec},h},h}}$. Recall that Lemma 4.2 was proved only in the stand alone settings, where in particular no interaction with a random system is considered. Since the proof of security of (Enc, Dec) is strongly-black-box, we have that Γ does not access, through interaction with $M^{\mathsf{A}_{\mathrm{KDI}}^{\mathrm{Breaker}^{\mathrm{Dec},h},h}}$, the function h. Therefore, Γ's answers are determined by the output behavior of $M^{\mathsf{A}_{\mathrm{KDI}}^{\mathrm{Breaker}^{\mathrm{Dec},h},h}}$ and the proof of Lemma 4.2 goes through also in this setting. Hence, Lemma 4.2 yields, letting $\delta(n) = \frac{1}{2p(n)}$, the existence of an efficient algorithm $\widetilde{M}^h$ with oracle access *only* to h, such that $\mathsf{Pr}_{h,r_{\mathsf{A}},r_\Gamma}[\widetilde{M}^h \leftrightarrow \Gamma(1^n) \text{ wins}] > \frac{1}{2p(n)}$ for infinitely many n's.

Our final step is to emulate $\widetilde{M}^h$, where rather than accessing h we randomly chooses the answer of each time one is requested (and cache it). The latter emulation breaks the cryptographic assumption with probability at least $\frac{1}{2p(n)}$ for infinitely many n's and since it is also efficient, it implies that Γ is not secure. □

5 Applying Our Technique to Other Primitives

It seems tempting to try and use the above Breaker also to show the impossibility of constructing other KDI-secure primitives. Consider for instance pseudorandom functions or permutations that are supposed to be secure even if the adversary can obtain its value on a function of its secret key. Halevi and Krawczyk [HK07] show that a deterministic construction cannot exist, but give a construction in case the permutation has an additional public parameter (i.e., *salt*) chosen after the challenge function is fixed. Their construction, however, compresses (e.g., maps n bits to $n/2$).

It is indeed possible to generalize our techniques to this case, as long for as the pseudorandom functions are injective for *every key*. In this case, Breaker finds a key k such that $f^{k,r}(h(k)) = c$, where f is the pseudorandom function and r is the random salt. The reason this method fails if the construction compresses (as the one given by Halevi and Krawczyk [HK07]), is that Breaker as defined above does not seem to give useful information about the key anymore, since it is unlikely that the correct key is the lexicographically smallest.

It seems that we also cannot utilize our Breaker for the general case of length increasing (non-injective) pseudorandom functions (or equivalently, for the case that we are allowed to make several KDI queries). Consider the question whether a given pseudorandom function is constant on a negligible fraction of the keys (e.g., on a single key k it holds that $f^{k,r}(\cdot) := 0^\ell$). Deciding whether a given function has this property or not might be infeasible. Yet, using for instance the Breaker of Section 4, we can easily find the right answer: ask the Breaker on $(h, 0^\ell)$, where h is a random hash function, and answer "Yes" is the Breaker finds some consistent key. Thus, in this setting our Breaker gives us an extra power that we cannot emulate.

Acknowledgments

We are very grateful to Oded Goldreich, Jonathan Hoch, Gil Segev, Omer Reingold and Udi Wieder for useful discussions. We thank the anonymous referees for many useful comments.

References

[AR02] Martín Abadi and Phillip Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). *Journal of Cryptology*, 15(2):103–127, 2002.

[Bar01] Boaz Barak. How to go beyond the black-box simulation barrier. In *Proceedings of the 42nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 106–115. IEEE Computer Society, 2001.

[BHHO08] Dan Boneh, Shai Halevi, Michael Hamburg, and Rafail Ostrovsky. Circular-secure encryption from decision Diffie-Hellman. In *Advances in Cryptology – CRYPTO 2008*, 2008.

[BRS02] John Black, Phillip Rogaway, and Thomas Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In Kaisa Nyberg and Howard M. Heys, editors, *Selected Areas in Cryptography*, volume 2595 of *Lecture Notes in Computer Science*, pages 62–75, 2002.

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

[CW79] J. Lawrence Carter and Mark N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18(2):143–154, April 1979.

[DOP05] Yevgeniy Dodis, Roberto Oliveira, and Krzysztof Pietrzak. On the generic insecurity of the full domain hash. In *Advances in Cryptology – CRYPTO 2005*, 2005.

[DY83] Danny Dolev and Andrew C. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, 1983.

[GGKT05] Rosario Gennaro, Yael Gertner, Jonathan Katz, and Luca Trevisan. Bounds on the efficiency of generic cryptographic constructions. *SIAM Journal on Computing*, 35(1):217–246, 2005.

[GKM+00] Yael Gertner, Sampath Kannan, Tal Malkin, Omer Reingold, and Mahesh Viswanathan. The relationship between public key encryption and oblivious transfer. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science*, 2000.

[GT00] Rosario Gennaro and Luca Trevisan. Lower bounds on the efficiency of generic cryptographic constructions. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FOCS)*, 2000.

[HHRS07] Iftach Haitner, Jonathan J. Hoch, Omer Reingold, and Gil Segev. Finding collisions in interactive protocols – A tight lower bound on the round complexity of statistically-hiding commitments. In *Proceedings of the 47th Annual Symposium on Foundations of Computer Science (FOCS)*, 2007.

[HK05] Omer Horvitz and Jonathan Katz. Bounds on the efficiency of "black-box" commitment schemes. In *ICALP '05*, pages 128–139, 2005.

[HK07] Shai Halevi and Hugo Krawczyk. Security under key-dependent inputs. In Sabrina De Capitani di Vimercati and Paul Syverson, editors, *14th ACM conference on Computer and communications security*, pages 466–475, 2007.

[Hof08] Dennis Hofheinz. Possibility and impossibility results for selective decommitments. Technical Report 2008/168, eprint.iacr.org, April 2008.

[HU08] Dennis Hofheinz and Dominique Unruh. Towards key-dependent message security in the standard model. In Nigel Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, 2008.

[IJK07] Russell Impagliazzo, Ragesh Jaiswal, and Valentine Kabanets. Chernoff-type direct product theorems. In *Advances in Cryptology – CRYPTO 2007*, pages 500–516, 2007.

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

[KST99] Jeong Han Kim, Daniel R. Simon, and Prasad Tetali. Limits on the efficiency of one-way permutation-based hash functions. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 535–542, 1999.

[RTV04] Omer Reingold, Luca Trevisan, and Salil P. Vadhan. Notions of reducibility between cryptographic primitives. In *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004*, volume 2951 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2004.

[Rud88] Steven Rudich. *Limits on the Provable Consequences of One-Way Functions*. PhD thesis, U.C. Berkeley, 1988.

[Sim98] Daniel Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In *Advances in Cryptology – EUROCRYPT '98*, volume 1403 of *Lecture Notes in Computer Science*, pages 334–345. Springer, 1998.

[Wee07] Hoeteck Wee. One-way permutations, interactive hashing and statistically hiding commitments. In *Theory of Cryptography, Fourth Theory of Cryptography Conference, TCC 2007*, pages 419–433, 2007.

A Gennaro-Trevisan Style Proof for Lemma 3.4

In this section we prove an alternative non-uniform version of Lemma 3.4.

Lemma A.1 (non-uniform version of Lemma 3.4)**.** *Let* A *be a non-uniform adversary that gets* h *as an auxiliary input and has oracle access to* π *and* $\text{Breaker}^{\text{Dec}^\pi,h}$*. Assume that* A *and* Dec *satisfy the upper bounds* $\mu_{\mathsf{A}}(n) = \mu_{\text{Dec}}(n) = n^C$ *for some* $C \in \mathbb{N}$*. For* $\varepsilon(n) = 2^{-n^{1/(2C)}}$ *we have that*

$$\mathsf{Pr}_{h\leftarrow\mathcal{H},\pi\leftarrow\Pi,y\leftarrow\{0,1\}^n}[\mathsf{A}^{(\pi,\text{Breaker}^{\text{Dec}^\pi,h})}(y,h) = \pi^{-1}(y)] < 3\varepsilon \ .$$

Our main tool is the following lemma.

Lemma A.2. *Let* A *be a non-uniform adversary that gets* h *as an auxiliary input, and has oracle access to* π *and* Breaker*, and assume that*

$$\mathsf{Pr}_{y\leftarrow\{0,1\}^n}\left[\mathsf{A}^{(\pi,\text{Breaker}^{\text{Dec}^\pi,h})}(y,h) = \pi^{-1}(y) \wedge \overline{\text{Bad}(y)}\right] > \varepsilon(n) \ ,$$

where $\text{Bad}(y)$ *is the event that* $\mathsf{A}(y,h)$ *makes a query* $\text{Breaker}^{\text{Dec}^\pi,h}(t,c)$ *for which* $|\text{Diff}^\pi(t,c,h,y)| \geq \mu_{\text{Dec}}(\mu_{\mathsf{A}}(n))^2$*. Then,* π *can be described using* $\log((2^n - s(n))!) + 2s(n)\log\left(\frac{e2^n}{s(n)}\right) + \mu_{\mathsf{A}}(n)^2\mu_{\text{Dec}}(\mu_{\mathsf{A}}(n))$ *bits, where* $s(n) = \varepsilon(n)2^n/\left(2\mu_{\mathsf{A}}(n)(\mu_{\text{Dec}}(\mu_{\mathsf{A}}(n)))^2\right)$*.*

Before proving Lemma A.2, let us first use it for proving Lemma A.1.

Proof. (of Lemma A.1) Claim 3.7 guarantees that for every value of π and y it holds that $\mathsf{Pr}_{h\leftarrow\mathcal{H}}[\text{Bad}(y)] < \mu_{\mathsf{A}}(n)2^{-\mu^{-1}_{\text{Dec}}(n)} = n^C 2^{-n^{1/C}} \leq 2^{-n^{1/2C}} = \varepsilon(n)$. It follows that

$$\begin{aligned}
&\mathsf{Pr}_{h\leftarrow\mathcal{H},\pi\leftarrow\Pi,y\leftarrow\{0,1\}^n}[\mathsf{A}^{(\pi,\text{Breaker}^{\text{Dec}^\pi,h})}(y,h) = \pi^{-1}(y)] \\
&\leq \mathsf{Pr}_{h\leftarrow\mathcal{H},\pi\leftarrow\Pi,y\leftarrow\{0,1\}^n}[\mathsf{A}^{(\pi,\text{Breaker}^{\text{Dec}^\pi,h})}(y,h) = \pi^{-1}(y) \wedge \overline{\text{Bad}(y)}] + \mathsf{Pr}_{h\leftarrow\mathcal{H},\pi\leftarrow\Pi,y\leftarrow\{0,1\}^n}[\text{Bad}(y)] \\
&\leq \mathsf{Pr}_{h\leftarrow\mathcal{H},\pi\leftarrow\Pi,y\leftarrow\{0,1\}^n}[\mathsf{A}^{(\pi,\text{Breaker}^{\text{Dec}^\pi,h})}(y,h) = \pi^{-1}(y) \wedge \overline{\text{Bad}(y)}] + \varepsilon
\end{aligned}$$

Lemma A.2 implies that for any h, the fraction of oracles for which $\mathsf{Pr}_{y\leftarrow\{0,1\}^n}[\mathsf{A}^{(\pi,\mathrm{Breaker}^{\mathrm{Dec}^\pi,h})}(y,h) = \pi^{-1}(y) \wedge \overline{\mathrm{Bad}(y)}] < \varepsilon$ is at most

$$\begin{aligned}\frac{(2^n-s(n))!\left(\frac{e2^n}{s(n)}\right)^{s(n)}2^{\mu_{\mathsf{A}}(n)^2\mu_{\mathrm{Dec}}(\mu_{\mathsf{A}}(n))}}{(2^n)!} &\le \frac{\left(\frac{e2^n}{s(n)}\right)^{s(n)}2^{\mu_{\mathsf{A}}(n)^2\mu_{\mathrm{Dec}}(\mu_{\mathsf{A}}(n))}}{(2^n-s(n))^{s(n)}}\\ &= \Big(\frac{e2^n}{s(n)\underbrace{(2^n-s(n))}_{\ge 2^{n-1}}}\Big)^{s(n)}2^{\mu_{\mathsf{A}}(n)^2\mu_{\mathrm{Dec}}(\mu_{\mathsf{A}}(n))}\\ &\le \Big(\frac{2e}{s(n)}\Big)^{s(n)}2^{\mu_{\mathsf{A}}(n)^2\mu_{\mathrm{Dec}}(\mu_{\mathsf{A}}(n))} \le 2^{-2^{n/2}} < \varepsilon\ ,\end{aligned}$$

which implies the lemma. □

Proof. (of Lemma A.2) We assume without loss of generality that $\varepsilon(n) > 2\mu_{\mathsf{A}}(n)2^{-n}$, since otherwise the statement is trivial. Our description of π consists of the following parts: the description of a set $S \subseteq \{0,1\}^n$, the description of the image of S under π (which roughly corresponds to the y's on which A succeeds in inverting), the description of the permutation that π implies if restricted on $\{0,1\}^n \setminus S$ (i.e., the elements not in S) and finally the description of $\{h_m \in h \mid m \le \mu_{\mathsf{A}}(n)\}$.

The description of S and the image of S both require $\log\binom{2^n}{|S|} \le |S|\log(e\frac{2^n}{|S|})$ bits. The description of the permutation requires at most $\log((2^n-|S|)!)$ bits. To store the functions h_m takes $\mu_{\mathsf{A}}(n)^2\mu_{\mathrm{Dec}}(\mu_{\mathsf{A}}(n))$ bits, for some appropriate family $\mathcal{H}$. Thus, in total $\log((2^n-|S|)!) + 2|S|\log(e\frac{2^n}{|S|}) + \mu_{\mathsf{A}}(n)^2\mu_{\mathrm{Dec}}(\mu_{\mathsf{A}}(n))$ bits are sufficient. In the following we succeed in making S as big as $|S| = \varepsilon(n)2^n/\big(2\mu_{\mathsf{A}}(n)(\mu_{\mathrm{Dec}}(\mu_{\mathsf{A}}(n)))^2\big)$, which implies our upper bound on the description size of π.

Defining the set S. We use the following, inefficient, algorithm to create S: we start by letting $I = \left\{y \in \{0,1\}^n \,:\, \mathsf{A}(y) = \pi^{-1}(y) \wedge \overline{\mathrm{Bad}(y)}\right\}$ and iteratively do the following. First, remove the lexicographic smallest element y from I and add $\pi^{-1}(y)$ to S. Next, emulate $\mathsf{A}^{(\pi,\mathrm{Breaker}^{\mathrm{Dec}^\pi,h})}(y,h)$ and remove all queries A makes whose answers are in I from I (without putting them into S). In addition, for each query to $\mathrm{Breaker}^{\mathrm{Dec}^\pi,h}(t,c)$ done by $\mathsf{A}^{(\pi,\mathrm{Breaker}^{\mathrm{Dec}^\pi,h})}(y,h)$, remove the images $\pi(x)$ from I for all $x^* \in \mathrm{Diff}^\pi(t,c,h,y)$ (note that y itself is not removed from I, as we already removed it). Once the emulation is over, repeat with the next element. Since for every emulation we remove at most $\mu_{\mathsf{A}}(n)+\mu_{\mathsf{A}}(n)\cdot\mu_{\mathrm{Dec}}(\mu_{\mathsf{A}}(n))^2$ elements from I before moving another element to S, we have that $|S| \ge \varepsilon(n)2^n/2\mu_{\mathsf{A}}(n)(\mu_{\mathrm{Dec}}(\mu_{\mathsf{A}}(n)))^2$.

The reconstruction of π. We now show that we can reconstruct π from the given information. For this, we first reconstruct the oracle outside of S from the given information. Then pick the lexicographic smallest element $y \in S$ whose preimage is not yet known, and emulate $\mathsf{A}^{\pi,\mathrm{Breaker}^{\mathrm{Dec}^\pi,h}}(y,h)$. We first consider the queries $\pi(x)$ done by $\mathsf{A}(y,h)$. The definition of S yields that we either know the answer for this query, or we are guaranteed that $\pi(x) = y$ (and we can stop the emulation). So it is left to consider the queries $\mathrm{Breaker}^{\mathrm{Dec}^\pi,h}(t,c)$. We note the that if $k \in \{0,1\}^t$ is the value we should return as the answer of $\mathrm{Breaker}^{\mathrm{Dec}^\pi,h}(t,c)$, then the answers to all π-queries made by $\mathrm{Breaker}^{\mathrm{Dec}^\pi,h}(t,c)$ when it calls $\mathrm{Dec}^\pi(k,c)$ (see Algorithm 3.1) should be known, where the only exception is a query on $\pi^{-1}(y)$ if such occurs. Therefore, we try all candidates x^* (i.e., the elements whose image we don't know at this point) for $\pi^{-1}(y)$, and emulate $\mathrm{Dec}^{\pi|x^*\to y}(k,c)$. The latter emulation succeeds unless a query is made whose answer we don't know. In this case, we know by the above observation that the current pair (k,x^*) is not the one we are looking for, and we can safely move to the next candidate for x^*. Finally, note that if a successful emulation of $\mathrm{Dec}^{\pi|x^*\to y}(k,c) = h(k)$ done by $\mathrm{Breaker}^{\mathrm{Dec}^\pi,h}(t,c)$ satisfies $\mathrm{Dec}^{\pi|x^*\to y}(k,c) = h(k)$, then k must be the correct answer to $\mathrm{Breaker}^{\mathrm{Dec}^\pi,h}(t,c)$. The reason for that x^* would be in $\mathrm{Diff}^\pi(t,c,h,y)$, and therefore cannot be in S. All in all, we can emulate $\mathsf{A}(y,h)$'s run correctly and obtain the correct $\pi^{-1}(y)$ as the output of A. □