# Security under Key-Dependent Inputs

**Abstract**

In this work we re-visit the question of how to build cryptographic primitives that remain secure even when queried on inputs that depend on the secret key. This was investigated by Black et al. in the context of randomized encryption schemes (in the random oracle model), and we extend the investigation to deterministic schemes (such as PRFs and block ciphers) and to the standard model. We term this notion "*security against key-dependent-input attack*", or KDI-security for short. Our motivation for studying KDI security is the existence of significant real-world implementations of "deterministic encryption" (in the context of storage encryption) that actually rely on their building blocks to be KDI secure.

We consider many natural constructions for PRFs, ciphers, tweakable ciphers and even randomized encryption, and examine them with respect to their KDI security. We exhibit limitation of this notion by demonstrating that many natural constructions fail to be KDI secure, and also study the limited cases where some measure of KDI security can be provably achieved.

## 1 Introduction

Does it make sense for an application to self-encrypt an encryption key? That is, if $E_s$ represents an encryption function with key $s$, would it ever be the case that an application needs to store or transmit $E_s(s)$? Cryptographers typically see this as a *dangerous abuse* of an encryption scheme, and standard security criteria for encryption scheme do not take this possibility into account. Still, there are applications where such form of security is helpful. This security concern was defined and studied by Black et al. [4] under the name KDM-security (for Key-Dependent-Messages). It was shown in [4] that KDM-security can be achieved in the random-oracle model, and later Boneh and Ostrovsky observed that the Cramer-Shoup cryptosystem can be proven KDM-secure in the standard model [6].

If "encrypting your own key" is somewhat abusive for randomized encryption, using this practice with deterministic constructions (such as pseudorandom functions and permutations) seems even more dangerous. The present work was motivated by a significant real-world application that turned out to be doing just that: An IEEE standard group was developing a standard for "sector level encryption" [10], which must be length-preserving and hence deterministic. The group was considering a transformation based on the tweakable cipher of Liskov et al. [11], but some members objected, citing an attack that can be mounted when the proposed transformation is used to transform its own secret key. An argument ensued as to whether or not this is a real problem or just a curiosity that would never happen in the real world. The argument was decided when the group was informed that the implementation of disk encryption in Windows Vista™ can store to the disk an encryption of its own secret keys in some situations. Consequently, the group switched to a different transformation, based on Rogaway's work [14], for which the particular attack in question does not seem to apply (see more details in Section 4 and [1]).

In this work we re-visit KDM security with a focus on deterministic constructions. We rename the notion to KDI security, to stress that we are not talking just about encryption (and hence the

Input is not necessarily a Message). We examine constructions of pseudorandom functions and (tweakable) pseudorandom permutations with respect to this notion, both in the standard model and in the "ideal cipher model".

Not surprisingly, KDI security seems even harder to achieve for deterministic constructions than for randomized ones. In particular, we observe that KDI-security of deterministic schemes cannot be achieved even in the "ideal cipher model" without restricting the key-dependent queries that the attacker can make. Specifically, similarly to the setting of "related key attacks" [3], allowing the attacker to query a function $f_s$ on multiple functions of the key necessarily translates into a KDI attack that *recovers the full key*, and this attack works even if the underlying primitive is an "ideal cipher" or a random oracle! In practical terms this means that an application that allows processing with a secret key unrestricted multiple forms of the same key is necessarily insecure.

For this reason, we parametrize KDI security by the set of functions of the key that the attacker can use in its queries. Given the impossibility mentioned above when the set of functions is too rich, it makes sense to investigate the modest notion of KDI-security with respect to just a single function, as we may at least hope that even an abusive implementation that "encrypts its own key" will only do so in one form, rather than "encrypting" many copies of the key in many different forms.

In this light, we study the question of a deterministic scheme that is KDI-secure with respect all efficient functions of the key, as long as the attacker is restricted to query a single function of its choice in the attack. We show that even this modest goal cannot be achieve, since for each PRF we can construct a function of the key for which the given PRF breaks. This last result, however, uses functions of the key that depend on the construction in question. (Indeed, we show that in the "ideal cipher model" it is possible to get constructions which are KDI secure with respect to any function that does not depend on the ideal cipher itself, e.g., one can query the cipher $E_s$ on the key $s$ itself or on some of its bits, but not on the function $g(s) = E_s(0)$.)

Seeing that we cannot have a single construction that is secure against any function of the key, we consider the very minimalistic goal of having a different secure construction for each function of the key. That is, can we at least describe a transformation that given a function $g$ produces a PRF $F^{(g)}$ which is KDI secure with respect to $g$ (i.e., remains secure when the attacker learns $F_s^{(g)}(g(s))$)? It turns out that even this goal is not easy to achieve in the standard model, and we only exhibit a partial positive answer. Specifically, we describe two (somewhat contrived) constructions, one that works for functions $g$ where $g(s)$ has high min-entropy (for a random $s$), and another that works when $g(s)$ has low Shannon entropy. Exhibiting a construction that works for functions where neither condition holds (or a construction in the case where we do not know if $g$ is one or the other) remains an open problem. On the positive side, we also show an arguably non-contrived construction that is KDI-secure with respect to the identity function (i.e., remains secure even when the attacker sees $f_s(s)$).

**Additional results.** Having demonstrated that KDI security for deterministic constructions can only be achieved in a very partial manner in the standard model, we return to the "ideal cipher model" to study the KDI security of tweakable ciphers (which were the initial motivation of this work). We establish a definition of KDI security for tweakable cipher, describe the attack on the scheme from [11] thus demonstrating that it is not KDI-secure (even in the "ideal cipher model"), and then show that some other schemes (including the one from [14]) are KDI-secure in this model.

Finally, we take another look at randomized encryption, and consider the scheme that was proven secure by Black et al. in the random-oracle model. Specifically, they analyzed the encryption

scheme $\mathrm{Enc}_s(x) = (r, f_s(r) \oplus x)$ where $f$ is implemented using the random oracle $f_s(x) = H(s|x)$, whereas we want to study this scheme in the standard model when $f_s$ is modeled as a PRF. We show that this constructions is not necessarily secure in the standard model even with respect to the identity function, and more surprisingly it fails even for "natural" choices of the PRF $f_s$, such as when instantiated using the Davies-Meyer construction.

**The moral.** The multiple negative results in this work lend strong support to the "common cryptographic wisdom" that the practice of self encryption of a key is a dangerous abuse of a cryptosystem. We demonstrate that many security goals that can be stated with respect to this practice inherently cannot be achieved, and even more damaging the failures are sometimes manifested even in the idealized models or with respect to very natural constructions. Our counter-example for the case of randomized encryption is particularly troubling, as we show a failure of a textbook construction for symmetric encryption with respect to a very natural implementation of its components.

We have also a few positive results, showing that some constructions can achieve some notion of KDI security (and sometimes this is doable even in the standard model). Two interesting open questions that remain are to (a) find a fast randomized symmetric encryption scheme (that does not use "public key tools"), which is KDI-secure for "interesting" functions of the key (at least the identity function); and (b) find, for every deterministic function $g(s)$ a PRF/PRP that is KDI-secure with respect to the function $g$.

## 2 Definitional approach and some intrinsic limitations

Roughly, to define security with respect to key-dependent input attacks we modify the standard attack scenarios for the various primitives that we study by allowing the attacker to query its oracles not only on explicit (i.e., key-independent) strings, but also on functions of the secret key. That is, where the original notion provided the attacker access to an oracle $O(\cdot)$, we add an oracle $O'(\cdot)$ that gets as input a description of a function $g$ (e.g., in the form of a circuit that computes the function) and outputs $O(g(s))$ where $s$ is the secret key of the construction in question. We will refer to the queries to $O'$ as *functional queries*. We extend this definitional approach to the "ideal cipher model" by allowing oracle access to keyed random permutations (and their inverses) and by possibly allowing the functional queries to depend on these oracles.

Ideally, we would like to find constructions that remain secure even when the attacker can query the primitive on any efficient function of the key. There are, however, some inherent limitations to this approach. For example, letting the attacker query a cipher $E_s$ on input $g(s) = E_s^{-1}(s)$, the key would be obviously exposed. A more general limitation arises in the context of deterministic primitives, as we show next.

**KDI-insecurity against unrestricted key-dependent queries.** The idea of this argument is that an attacker can try to apply many different functions to the key $s$, and use collisions of the form $g(s) = g'(s)$ to do a binary search for the key $s$. That is, the attacker uses two different functional queries $g, g'$, and checks if it gets the same answer on both. This (in essence) tells the attacker whether $g(s) = g'(s)$, which cuts the key-space by two. Here we describe a simple example of this argument, which is essentially the same as the one described in [3] in the context of related-key attacks.

Let $\Psi$ be any deterministic construction that has a secret key (such that the disclosure of $s$ compromises the security of $\Psi$). For simplicity (this is not essential for the general argument),

assume that both the key space and the input space of $\Psi$ is $\{0,1\}^n$, and that for every fixed key $s \in \{0,1\}^n$ the function $\Psi_s(\cdot)$ is injective. Consider now a set of functions $\{g_i, g'_i : 1 \leq i \leq n\}$ with the property that for every $i$ and every $s$ we have $g_i(s) = g'_i(s)$ if and only if the $i$'th bit of $s$ is zero. (An example is a set of functions containing additions of constants modulo $2^n$ as well as xor with constants from $\{0,1\}^n$. For example, for all $i < n$ we set $g_i$ to be xor with $0^{n-i}10^{i-1}$ and $g'_i$ can be addition of the same constant modulo $2^n$.)

The attacker then simply queries its oracle $\Psi'$ on the inputs $g_i$ and $g'_i$ for all $i$. If the $i$'th bit of the secret key is 0 then $g_i(s) = g'_i(s)$ and therefore $\Psi'(g) = \Psi(g_i(s)) = \Psi(g'_i(s)) = \Psi'(g')$ (because $\Psi$ is deterministic). On the other hand, if the $i$'th bit of the secret key is 1 then $g_i(s) \neq g'_i(s)$ and since $\Psi$ is injective it follows that also $\Psi'(g) = \Psi(g_i(s)) \neq \Psi(g'_i(s)) = \Psi'(g')$. The attacker can therefore determine all the bits of the secret key $s$ in violation of the security of $\Psi$.

**Parametrized definition.** As a consequence of the above observations, and similar to the case of key-related attacks [3], the definition of KDI-security will be parametrized by a class of function descriptions $\mathcal{C}$, and all the queries to the $O'$ oracle will be restricted to functions from $\mathcal{C}$. The question of whether KDI security with respect to a certain class $\mathcal{C}$ provides a meaningful level of security depends heavily on the application. In some cases anything less than "all polynomial-size circuits" may be insufficient while in others having $\mathcal{C}$ restricted to the identity function only (i.e., one is allowed to query the primitive on the key itself but not on other functions of the key) may suffice.

In many cases, providing security assurance against one function of the key, i.e., the case where $|\mathcal{C}| = 1$, will be of significant value: we may at least hope that even an abusive implementation that "encrypts its own key" will only do so in one form, rather than encrypting many copies of the key in many different forms. Given the limitations discussed above (and more to be shown in the sequel) we will judge different constructions under the "modest" requirement that they resist singleton classes $|\mathcal{C}| = 1$. We would like to get a construction that is KDI secure against *all* singleton classes (i.e., the attacker is allowed to choose a single function $g(s)$ to query but the function $g$ could be any efficient function of $s$). Unfortunately, examples such as the one with the function $g(s) = E_s^{-1}(s)$ demonstrate that even this modest goal cannot always be achieved. In such a case we will study the "minimalist" requirement that a construction is KDI secure against *one specific function* (and as we show in Section 3.2.1, even this is not easily achievable in the standard model).

# 3 Pseudorandom Functions

Below we use the convention that for security parameter $k$, the key for a pseudorandom function is a random $k$-bit string, and that the function is from $\{0,1\}^{\ell_{\text{in}}(k)}$ to $\{0,1\}^{\ell_{\text{out}}(k)}$ where $\ell_{\text{in}}$ and $\ell_{\text{out}}$ are efficiently computable and polynomially bounded. Then a family of pseudorandom functions is an ensemble

$$\mathcal{F} = \left\{ f_s : \{0,1\}^{\ell_{\text{in}}(k)} \to \{0,1\}^{\ell_{\text{out}}(k)} \;\middle|\; s \in \{0,1\}^k \right\}_{k \in \mathcal{N}}$$

and we require that there is an efficient evaluation procedure that given any $s \in \{0,1\}^k$ and any $x \in \{0,1\}^{\ell_{\text{in}}(k)}$ computes $y = f_s(x)$.

The standard security definition for pseudorandom functions as defined in [8] asserts that no feasible attacker $\mathcal{A}^{\phi}(1^k)$ (with oracle access to $\phi$) can distinguish with any significant advantage the case where $\phi = f_s$ for a random $s \in_R \{0,1\}^k$ from the case where $\phi$ is chosen as a random function from $\{0,1\}^{\ell_{\text{in}}(k)}$ to $\{0,1\}^{\ell_{\text{out}}(k)}$.

Following the rationale presented in Section 2, in order to capture KDI security of pseudorandom functions, we augment the standard definition of pseudorandom functions by letting the adversary also access another oracle $\phi'$ that takes as input a description of a function $g$ and outputs $\phi(g(s))$.

**Definition 1 (KDI-secure PRFs)** *A family $\mathcal{F}$ of pseudorandom functions is* KDI-secure *with respect to a class $\mathcal{C}$ of circuits if no feasible attacker $\mathcal{A}^{\phi,\phi'}(1^k)$ (with oracle access to $\phi, \phi'$) can distinguish with any significant advantage between the following two cases:*

1. *$\phi = f_s$ for a random $s \in_R \{0,1\}^k$ and $\phi'(g) = \phi(g(s))$ for any $g \in \mathcal{C}$;*
2. *$\phi$ is chosen as a random function $\phi : \{0,1\}^{\ell_{\text{in}}(k)} \to \{0,1\}^{\ell_{\text{out}}(k)}$, $s$ is chosen at random in $\{0,1\}^k$, and $\phi'(g) = \phi(g(s))$ for any $g \in \mathcal{C}$.*

**On KDI-insecure PRFs.** We first observe that secure PRFs (or block ciphers) are not necessarily KDI-secure, not even with respect to the identity function. Indeed, given any secure PRF family $F = \{F_s\}$, one can trivially modify it as follows: $F'_s(x) = s$ if $x = s$ and $F'_s(x) = F_s(x)$ otherwise. Clearly, the family $F' = \{F'_s\}$ is still a secure PRF, but it is not KDI-secure with respect to the identity function. Similarly, if we start with a secure cipher $E$ (a strong pseudorandom permutation) we can build another secure cipher $E'$ that is not KDI-secure with respect to the identity function:

$$E'_s(x) = \begin{cases} s & \text{if } x = s \\ E_s(s) & \text{if } x = E_s^{-1}(s) \\ E_s(x) & \text{otherwise} \end{cases}$$

### 3.1 Constructions in the "ideal-cipher model"

We saw above that the construction $f_s(x) = E_s(x)$ where $E$ is a secure block cipher is not necessarily KDI-secure. Here we show that this construction is at least KDI secure in the "ideal-cipher model". We begin by adapting our definition of KDI security to Shannon's "ideal cipher model".

Recall that in the ideal-cipher model, all the parties (including the attacker) are given black-box access to two tables $\Pi(\cdot,\cdot)$ and $\Pi^{-1}(\cdot,\cdot)$. These tables are chosen at random subject to the condition that for every "key" $s$, $\Pi(s,\cdot)$ is a permutation and $\Pi^{-1}(s,\cdot)$ is its inverse (and all these permutations are over the same domain). For simplicity of presentation we assume that on security parameter $k$, the key that selects the permutation is of length $k$ bits and the permutations themselves are over $\{0,1\}^k$. Namely, for each $s \in \{0,1\}^k$, $\Pi(s,\cdot)$ is a random permutation over $\{0,1\}^k$ , and $\Pi^{-1}(s,\cdot)$ is the inverse permutation.

We augment the definition of KDI-security to the ideal-cipher model by providing the attacker with oracle-access to $\Pi, \Pi^{-1}$, and more importantly by potentially allowing the class of function-descriptions in $\mathcal{C}$ to depend on $\Pi$ and/or $\Pi^{-1}$. Specifically, in this case we allow the circuits in $\mathcal{C}$ to include also $\Pi$-gates that on input $(s,x)$ return $\Pi(s,x)$ (and similarly also $\Pi^{-1}$-gates). When stating a result in this paper in the context of the ideal cipher model we will specify whether we assume the functional queries $g(s)$ to depend or not in the oracles $\Pi$ and $\Pi^{-1}$.

Note that when adapting Definition 1 to the "ideal cipher model", the attacker's advantage is measured with respect to the probability distribution where for each $s \in \{0,1\}^k$, $\Pi(s,\cdot)$ is a random permutation over $\{0,1\}^k$ and $\Pi^{-1}(s,\cdot)$ is the inverse permutation.

*Remark.* The distinction between circuits that include $\Pi$ and $\Pi^{-1}$ gates and circuits that do not is one of the main reasons for using the "ideal cipher model" in the KDI context. Indeed, in some

cases we would like to argue that a cipher is KDI-secure with respect to any function $g$ that "does not depend on the cipher itself". This restriction is generally not well defined in the standard model but can be captured in the "ideal cipher model" by specifying that the function $g$ is described by a circuit that does *not* include $\Pi$ or $\Pi^{-1}$ gates.

**KDI-security of $f_s(x) = \Pi_s(x)$.** It is easy to see that even in the "ideal cipher model", we can find functions $g$ that depend on $\Pi$ such that the construction $f_s(x) = \Pi(s, x)$ is not KDI-secure with respect to $g$. For example, if we set $g(s) = \Pi^{-1}(s, s)$ then $f_s(g(s)) = \Pi(s, \Pi^{-1}(s, s)) = s$. However, we can show that this construction is KDI secure with respect to every function $g$ that does not depend on $\Pi$, specifically:

**Theorem 1** *Let $g$ be any Boolean circuit* with no $\Pi$-gates or $\Pi^{-1}$-gates. *Then the construction $f_s(x) = \Pi_s(x)$ is a KDI-secure pseudorandom function in the "ideal cipher model" with respect to the singleton class $\mathcal{C} = \{g\}$.*

A sketch of the proof is in the appendix. Similar claims can be made for many of the published PRP-to-PRF constructions in the literature, e.g., the truncation construction [9], the XOR construction [13], etc.[1]

### 3.2 Constructions in the standard model

We have shown that an ideal cipher is also KDI secure with respect to any function $g$ that does not depend on the cipher itself. On the other hand, we saw, in the examples following Definition 1, that in the standard model a secure cipher (or PRF) does not have to be KDI-secure, not even with respect to simple functions such as the identity function. Here, we investigate to what extent one can build KDI-secure schemes in the standard model. As we will see, one obstacle is the fact that in the standard model one cannot impose independence between a PRF (or cipher) scheme and the function $g$ as we did in Theorem 1. Indeed, we show that in the standard model one cannot get a single construction that is KDI secure with respect to every singleton class $\{g\}$.

**Theorem 2 (No single construction for all $g$.)** *There exists no pseudorandom function family that is KDI-secure with respect to $\{g\}$ for all functions $g$.*

**Proof** Let $F = \{F_s\}$ be a pseudorandom family. Define $g_F(s) = F_s(0)$, and we show that $F$ is not KDI-secure with respect to $\{g_F\}$. An attacker $\mathcal{A}$ queries its key-dependent oracle to obtain $a = F_s(g_F(s)) = F_s(F_s(0))$; then it queries the $F$-oracle on 0 to obtain $b = F_s(0)$; finally, it queries the $F$-oracle on $b$ to obtain $c = F_s(b)$. $\mathcal{A}$ outputs 1 if $a = c$ and 0 otherwise. Clearly, when the oracle $F$ is answered with the pseudorandom function $F_s$ then $a = c$ and $\mathcal{A}$ outputs 1 with probability 1, while if the $F$-oracle is random then $a$ and $c$ are independent random values and hence $\mathcal{A}$ outputs 1 only with small probability. ■

[1] For these constructions one gets KDI security in the "ideal cipher mode" but not necessarily KDI security "beyond the birthday bound". (Getting PRF security beyond the birthday bound has been the initial motivation for these constructions.)

### 3.2.1 A tailored construction for every function g

Since we cannot have a single construction that is KDI secure against every function $g$, it is natural to ask whether we can at least have for every $g$ a tailored construction that is KDI secure against only this function $g$. Below we provide a partial positive answer to this question.

FIRST TRY. We begin with a simple construction that at first glance looks as if it should work. Let $f = \{f_s\}$ be a pseudorandom function (with keys and inputs that are $n$-bit long), fix some function $g : \{0,1\}^n \to \{0,1\}^n$, and define a family $F^{(g)} = \{F_s^{(g)}\}$ with $n$-bit keys and $(n-1)$-bit input:

$$F_s^{(g)}(x) = \begin{cases} f_s(1|x) & \text{if } x \neq g(s) \\ f_s(0) & \text{if } x = g(s) \end{cases} \tag{1}$$

Perhaps surprisingly, we show that this construction does not always work:

**Lemma 1** *There are functions $g$ s.t. $F^{(g)}$ from Eq. (1) is not KDI-secure with respect to $\{g\}$.*

**Proof** Define $g(s) = f_s(0)$. We show a KDI-attacker $\mathcal{A}$ that distinguishes between $F^{(g)}$ and a random function. The attacker $\mathcal{A}$ queries its key-dependent oracle to receive $a = F(g(s))$, then queries the oracle $F$ on $a$ to get $b = F(a)$ and outputs 1 if $a = b$ and 0 otherwise.

When the $F$-oracle is a real one (i.e., instantiated with $F_s^{(g)}$) then we have $a = F_s^{(g)}(g(s)) = f_s(0) = g(s)$, and therefore also $b = F_s^{(g)}(a) = F_s^{(g)}(g(s)) = f_s(0)$, thus $\mathcal{A}$ outputs 1 with probability 1. On the other hand, if the $F$ oracle is a random function then $a$ is a random value and $b = F(a)$ is an independent random value, hence $a = b$ happens only with small probability. ■

The reason for this counter-example was that the attacker was able to compute $g(s)$ given access to $F$. Indeed, we show that when this is impossible then the construction is secure.

**Lemma 2** *The construction $F^{(g)}$ is a KDI-secure PRF with respect to the singleton class $\mathcal{C} = \{g\}$, provided that the family $f$ is a secure PRF and has the property that for a random $s$, $g(s)$ is unpredictable even given oracle access to $f_s(\cdot)$.*

The proof is sketched in the appendix.

HIGH-ENTROPY $g$'S. Given Lemma 2, we may try to first design a PRF $f^{(g)}$ such that $g(s)$ is unpredictable given oracle access to $f_s^{(g)}$, and then use $f^{(g)}$ in the construction from Eq. (1). Indeed, if $g$ is a permutation then the construction $f_s^{(g)}(x) = f_{g(s)}(x)$ has the needed properties.

If $g$ is not a permutation but $g(s)$ still has enough min-entropy, then we can use a strong extractor $\mathsf{ext}$ and fix some public random seed, $r$, for it. Then $\mathsf{ext}_r(g(s))$ is close to random (even given $r$), so we can almost use it as the key for $f$. The only problem is that it is shorter than $n$ bits, so we use a pseudorandom generator to expand it to $n$ bits. This gives us the following randomized construction:

$$f_s^{(g,r)}(x) = f_{G(\mathsf{ext}_r(g(s)))}(x), \text{ and } F1_s^{(g,r)}(x) = \begin{cases} f_s^{(g,r)}(1|x) & \text{if } x \neq g(s) \\ f_s^{(g,r)}(0) & \text{if } x = g(s) \end{cases} \tag{2}$$

**Lemma 3** *If $f_s$ is a PRF, $\mathsf{ext} : R \times \{0,1\}^n \to \{0,1\}^m$ a strong entropy extractor and $G : \{0,1\}^m \to \{0,1\}^n$ a PRG, and if the distribution of $g(s)$ has at least $m + \omega(\log n)$ bits of min-entropy, then with high probability over the choice of $r$, the construction $F1^{(g,r)}$ is KDI-secure with respect to $g$.*

**Proof Sketch** The proof follows from Lemma 2 and since $g(s)$ is unpredictable even given $r$ and access to $f_s^{(g,r)}$. ■

LOW-ENTROPY $g$'S. The construction from Eq. (2) works for functions $g$ such that $g(s)$ has high min entropy. On the other end of the scale, if $g(s)$ has only little entropy then we can use a different construction. Specifically, if $g(s)$ has very many pre-images, then we can extract entropy from $s$ itself, so that $\mathsf{ext}_r(s)$ is close to random even given $g(s)$ and $r$. Namely, we have the following construction:

$$F2_s^{(r)}(x) = f_{G(\mathsf{ext}_r(s))}(x) \tag{3}$$

**Lemma 4** *If $f_s$ is a PRF, $\mathsf{ext} : R \times \{0,1\}^n \to \{0,1\}^m$ a strong entropy extractor and $G : \{0,1\}^m \to \{0,1\}^n$ a PRG, and if $g$ has the property that $g(s)$ for a random $s$ has at least $2^{m+\omega(\log n)}$ pre-images (except perhaps with negligible probability), then with high probability over the choice of $r$, the construction $F2^{(r)}$ is KDI-secure with respect to $g$.*

**Proof Sketch** The proof follows from the fact that $F2_s^{(r)}$ is pseudorandom even when the attacker is given $g(s)$ and $r$ (since $\mathsf{ext}_r(s)$ is close to random given both of these values). ■

PUTTING IT TOGETHER. Unfortunately, the "high entropy" condition in Lemma 3 is not the complement of the "low entropy" condition in Lemma 4. (One talks about min-entropy and the other about a notion similar to Shannon entropy.) Still, if $g$ is a regular function then these two conditions complement each other. (Recall that a function $g$ is regular if for all $s, s'$, $g(s)$ and $g(s')$ have the same number of pre-images.) Hence we get

**Theorem 3** *If PRFs exist, then for any regular function $g$, there is a randomized PRF $F^g$ that is KDI secure with respect to $\{g\}$ with high probability.*

**Proof Sketch** If the pre-images of $g$ are larger than $2^{n/2}$ then we use the construction from Eq. (2), and if they are smaller than $2^{n/2}$ then we use the construction from Eq. (3). ■

### 3.2.2 A more "natural" construction with respect to the identity function

We end this section by describing a somewhat more "natural" construction of KDI-secure PRF with respect to the identity function. This construction is reminiscent of an unpublished "symmetric PRF" construction due to Barak [2].[2] Assume that we have a pseudorandom generator $G : \{0,1\}^n \to \{0,1\}^{2m}$, and denote the first $m$ bits of output of $G$ on seed $s$ by $G_1(s)$ and the last $m$ bits by $G_2(s)$. Assume further that $G$ has the extra property that it is hard to find two different seeds $s_1, s_2$ such that $G_2(s_1) = G_2(s_2)$.

We call a generator with this extra property a ***collision-resistant generator***. For example, one can verify that such a pseudorandom generator (with $m = n$) can be constructed from any one-way permutation $\rho$ over $m$ bits using the Blum-Micali construction [5]:

$$G(s) = b(s)\ b(\rho(s))\ b(\rho^2(s))\ \ldots\ b(\rho^{m-1}(s)) \mid \rho^m(s) \tag{4}$$

where $b(\cdot)$ is a hard-core predicate for $\rho(\cdot)$. It is also plausible that ad-hoc constructions such as $G(s) = AES_s(0)|AES_s(1)|AES_s(2)\cdots$ have this property, since we can make $m$ sufficiently larger

[2] A "symmetric PRF" is a function $f(x,y)$, which is a PRF both when $x$ is viewed as the key and $y$ as the input, as well as when $y$ is viewed as the key and $x$ as the input.

than $n$. Given a collision-resistant pseudorandom generator $G(\cdot)$ and a pseudorandom function $f_s(\cdot)$, we construct another pseudorandom function $F_s(\cdot)$ by setting

$$F_s(x) = f_{G_1(s)}\left(G_2(x)\right). \tag{5}$$

**Lemma 5** *The construction $F$ is KDI-secure with respect to the singleton class containing the identity function $\mathcal{C} = \{ID\}$, assuming that $f$ is a pseudorandom function and $G$ is a collision-resistant pseudorandom generator.*

**Proof Sketch** We first note that because of the collision-resistance of $G$ it is unlikely that the attacker will ever query $F$ on two inputs $x, x'$ such that $G_2(x) = G_2(x')$. Assuming that this does not happen, we consider a "hybrid game" where the attacker's query of the identity function (i.e., its query of $F_s(s)$) is answered by $f_{G_1(s)}(r)$ for a random $r$ instead of by $f_{G_1(s)}(G_2(s))$. Then we show that the advantage of the attacker in the hybrid game is close to its advantage in the real game (since $G$ is a PRG), and at the same time this advantage must be insignificant (since $f$ is a PRF). ■

We remark that although the identity function does not appear to be "hard wired" in the definition of $F$ from above, this construction is not secure in general against any other function. For example, consider the function $g(s) = s + 1$, and we show a PRF $f$ and a collision-resistant PRG $G$ such that the resulting $F$ is not KDI-secure with respect to $\mathcal{C} = \{g\}$.

Assume that we have a PRG $G(s) = G_1(s)|G_2(s)$ which is collision resistant with respect to both $G_1$ and $G_2$. (Again, it is plausible that an ad-hoc construction such as $G(s) = AES_s(0)|AES_s(1)|AES_s(2)\cdots$ has this property, since we can have $|G_1(s)| = |G_2(s)| \gg |s|$.) We then define a new PRG

$$\tilde{G}(s|b) = \tilde{G}_1(s|b) \mid \tilde{G}_2(s|b), \text{ where } \begin{cases} \text{if } b = 0 \text{ then } \tilde{G}_1(s|b) = G_1(s),\ \tilde{G}_2(s|b) = G_2(s)|b \\ \text{if } b = 1 \text{ then } \tilde{G}_1(s|b) = G_2(s),\ \tilde{G}_2(s|b) = G_1(s)|b \end{cases}$$

It is clear that $\tilde{G}$ is a collision-resistant PRG. Also, let $f$ be a PRF with input which is one-bit longer than the key, and we modify it as follows:

$$f'_s(x|b) = \begin{cases} 0 & \text{if } x = s \\ f_s(x|b) & \text{otherwise} \end{cases}$$

Again, clearly $f'$ is still a PRF. However the construction $F$ from Eq. (5) is NOT KDI-secure with respect to $g(s) = s + 1$, since for any key whose last bit is zero, $s' = s|0$, we get

$$F_{s'}(s' + 1) = F_{s|0}(s|1) = f'_{\tilde{G}_1(s|0)}(\tilde{G}_2(s|1)) = f'_{G_1(s)}(G_1(s)|1) = 0$$

# 4 Tweakable Pseudorandom Permutations

Recall that a *tweakable* cipher (or tweakable pseudorandom permutation) [11] has a key and two inputs: a plaintext and a tweak. Below we use the convention that for security parameter $k$, both the plaintext and the tweak are $k$-bit strings, and that the cipher key is of length $\{0,1\}^{\ell(k)}$ where $\ell$ is some polynomially bounded function (we often use $\ell(k) = k$ or $\ell(k) = 2k$ ). Formally a family of tweakable pseudorandom permutations is an ensemble

$$\mathcal{P} = \left\{ p_{s,t} \in \mathcal{S}\left(\{0,1\}^k\right) \;\middle|\; s \in \{0,1\}^{\ell(k)}, t \in \{0,1\}^k \right\}_{k \in \mathcal{N}}$$

where $\mathcal{S}$ denotes the symmetric group, and we require that there are efficient evaluation and inversion procedures that given any $s, t \in \{0,1\}^k$ and any $x \in \{0,1\}^{\ell(k)}$ compute $y = p_{s,t}(x)$ and $z = p_{s,t}^{-1}(x)$, respectively.

The standard security definition for strong tweakable pseudorandom permutations as defined in [12, 11] asserts that no feasible attacker $\mathcal{A}^{\pi,\pi'}(1^k)$ (with oracle access to $\pi, \pi^{-1}$) can distinguish with any significant advantage the case where for a random $s \in_R \{0,1\}^k$ we set $\pi(t,x) \equiv p_{s,t}(x)$ and $\pi^{-1}(t,x) \equiv p_{s,t}^{-1}(x)$, from the case where for every $t \in \{0,1\}^k$, $\pi(t,\cdot)$ is chosen as a random permutation over $\{0,1\}^{\ell(k)}$ and $\pi^{-1}(t,\cdot)$ is set to the inverse permutation.

Adding KDI-security to this definition requires some choices. The attacker in this model has two oracles, each with two inputs (namely $\pi(\cdot,\cdot)$ and $\pi^{-1}(\cdot,\cdot)$) and we need to decide what input to what oracle can depend on the key. In this work we only consider the variant where the plaintext/ciphertext inputs to both $\pi$ and $\pi^{-1}$ can depend on the key, but not the tweaks. The reason that we do not consider key-dependent tweaks is that tweaks typically represent some context information or label that comes from a higher layer (e.g., in the storage application that motivated this paper the tweak represents the physical position where the data is to be stored), and so it may be reasonable to assume that it does not depend on the key.

With these choices, we modify the standard definition of tweakable PRPs by giving the adversary access to two additional oracles $\psi, \psi^{-1}$ that take as input a tweak $t$ and a description of a function $g$ and output $\pi(t, g(s))$ and $\pi^{-1}(t, g(s))$, respectively.

**Definition 2 (KDI-secure tweakable strong PRPs)** *A family $\mathcal{P}$ of tweakable pseudorandom permutations is* KDI-secure *with respect to a class $\mathcal{C}$ of circuits if no feasible attacker $\mathcal{A}^{\pi,\pi^{-1},\psi,\psi^{-1}}(1^k)$ can distinguish with any significant advantage between the following two cases:*

1. *The key $s \in_R \{0,1\}^k$ is chosen at random, and for any $t, x, g$ the oracles are set as, $\pi(t,x) \equiv p_{s,t}(x)$, $\pi^{-1}(t,x) \equiv p_{s,t}^{-1}(x)$, $\psi(t,g) \equiv p_{s,t}(g(s))$, and $\psi^{-1}(t,g) \equiv p_{s,t}^{-1}(g(s))$;*
2. *The key $s \in_R \{0,1\}^k$ is chosen at random, for every $t \in \{0,1\}^k$ we set $\pi(t,\cdot)$ to a random permutation over $\{0,1\}^\ell$ and $\pi^{-1}(t,\cdot)$ to its inverse, and then $\psi(t,g) \equiv \pi(t,g(s))$, and $\psi^{-1}(t,g) \equiv \pi^{-1}(t,g(s))$.*

As before, Definition 2 is adapted to the "ideal cipher model" by giving oracle access to the ideal cipher $\Pi, \Pi^{-1}$ to the construction itself, the attacker $\mathcal{A}$, and potentially also the circuits in $\mathcal{C}$.

Below we demonstrate that some constructions of tweakable ciphers in the literature are KDI insecure against simple functions of the key, while others can be proven secure in the ideal cipher model.

**KDI-insecurity of the LRW constructions.** Consider the following instantiation of the second construction of Liskov et al. from [11]. This instantiation has two keys, denoted $s_1, s_2$, where $s_1$ is used as a key for an underlying block cipher $E$ and $s_2$ is treated as an element of $GF(2^k)$ with $k$ the block-size of $E$. This construction then defines the following tweakable cipher, with both the block size and the tweak size equals to $k$ bits:

$$\tilde{E}_{s_1,s_2}(t,x) = E_{s_1}((t \cdot s_2) \oplus x) \oplus (t \cdot s_2) \tag{6}$$

where $t \cdot s_2$ is a multiplication in $GF(2^k)$). In [11] it is shown that the generic construction $E_{s_1}(h_{s_2}(t) \oplus x) \oplus h_{s_2}(t)$ is a secure tweakable cipher when $E$ is a secure cipher and $h$ is a "xor-universal" hash function, which implies the security of Eq. (6) since $h_{s_2}(t) = t \cdot s_2$ is indeed xor-universal.

However, as pointed out when this construction was considered for the IEEE 1619 standard, this construction is not KDI-secure with respect to the function $g(s_1, s_2) = s_2$ (i.e., when "encrypting" the element $s_2$ from the secret key). The attacker can query $\psi(0, g)$ (i.e., using tweak value 0 and "plaintext" $s_2$) and also $\pi(1, 0)$ (i.e., tweak value 1 and "plaintext" 0), thus getting

$$c_1 = \tilde{E}_{s_1,s_2}(0, s_2) = E_{s_1}(s_2) \quad \text{and} \quad c_2 = \tilde{E}_{s_1,s_2}(1, 0) = E_{s_1}(s_2) \oplus s_2$$

Next the attacker can compute $s_2 = c_1 \oplus c_2$ and then verify this value (e.g., by asking to "decrypt" the value of $c_1 \oplus 2s_2$ with respect to the tweak value 2).

**KDI-security of the "trivial construction".** Alternatively, consider the "trivial" construction of tweakable SPRP from a block cipher

$$\tilde{E}_s(t, x) = E_{E_s(t)}(x) \tag{7}$$

It is easy to see that if $E$ is a secure cipher then this construction is a secure tweakable cipher. Although there are functions $g$ for which this construction is not KDI-secure (for example the function $g(s) = E^{-1}_{E_s(t)}(0)$, we can show, however, that $\tilde{E}$ from Eq. (7) is KDI-secure in the "ideal cipher model" with respect to any function that *does not depend on the cipher itself.*

**Lemma 6** *Let $g$ be any Boolean circuit* with no $\Pi$-gates or $\Pi^{-1}$-gates. *Then the construction $\tilde{E}$ from Eq. (7) is a KDI-secure tweakable strong pseudorandom permutation in the "ideal cipher model" with respect to the singleton class $\mathcal{C} = \{g\}$.*

**Proof Sketch** Again, the proof is straightforward. Recall that the adversary in this game has six oracles: $\Pi(\cdot,\cdot)$ and $\Pi^{-1}(\cdot,\cdot)$ that represent the ideal cipher, $\tilde{E}(\cdot,\cdot)$ and $\tilde{E}^{-1}(\cdot,\cdot)$ that represent either the construction with a fixed random key $s$ or a random tweakable permutation (independent of $\Pi$), and $\psi(\cdot)$ and $\psi^{-1}(\cdot)$ that allow key-dependent queries[3] with $\psi(t) = \tilde{E}(t, g(s))$ and $\psi^{-1}(t) = \tilde{E}^{-1}(t, g(s))$ (where $s$ is the secret key in case one and just a random string in case two).

Similarly to the proof of Theorem 1, the proof goes by arguing that the attacker is very unlikely to ever query its $\Pi$ or $\Pi^{-1}$ oracles on the right "key" $s$, and without such queries the view of the attacker is the same in both cases. ■

**Other constructions.** We comment that a similar lemma can be proven also for Rogaway's XEX construction from [14], where on input $x$ and tweak $(i, j)$ one computes:

$$\tilde{E}_s((i, j), x) = E_s((2^i \cdot E_s(j)) \oplus x) \oplus (2^i \cdot E_s(j)) \tag{8}$$

Namely, this construction too can be proven KDI-secure in the "ideal cipher model" with respect to any function $g$ that does not depend on the ideal cipher. The proof itself is very similar to Rogaways's proof of security for XEX [14]. The key-dependent queries are handled using the fact that in the "ideal cipher model" the quantity $E_s(j)$ is independent of $s$ for all $j$, and therefore the attacker is unlikely to be able to issue two queries for which $(2^i \cdot E_s(j)) \oplus x = (2^{i'} \cdot E_s(j')) \oplus x'$ (even if $x, x'$ can be set as functions of the secret key $s$).

[3] Compared to Definition 2 we slightly simplify notations here by having $\psi, \psi^{-1}$ as single-input oracles. We can do this because the function $g$ is always the same, since we are interested in the singleton class $\mathcal{C} = \{g\}$.

## 5 Symmetric Encryption

The question of KDI-security for encryption was studied by Black et al. [4] (under the name KDM-security). They presented a definition of security (which follows the same rationale as we discussed in Section 2), and proved that it can be easily met in the random-oracle model.

One thing that makes encryption schemes an easier case for KDI security than PRFs and tweakable ciphers is that they are randomized. Indeed, as opposed to the situation with deterministic constructions, for randomized encryption there exist fully KDI-secure constructions based on standard number-theoretic assumptions. Specifically, it was shown by Boneh and Ostrovsky [6] that the Cramer-Shoup encryption scheme [7] is "circularly secure" assuming the hardness of the decision Diffie-Hellman problem. Using our terminology this means that Cramer-Shoup is KDI-CCA-secure with respect to the class $\mathcal{C}$ of all deterministic poly-size circuits. (Indeed, [6] observed that the original proof of security from [7] proves also this stronger statement, since the simulator in that proof knows the secret key.) However, we do not know of a construction that achieves similar level of KDI security in the standard model without using "public key tools" (and in particular we do not know of a construction that can be implemented in speeds comparable to AES).

Below we confider a very natural PRF-based symmetric encryption scheme, which is (a slight simplification of) the scheme that was proven secure in the random-oracle model by Black et al. Specifically, We show that not only this construction fails to be KDI secure in the standard model, but this failure is manifested even for a natural instantiations of the PRF (specifically when instantiated with a Davies-Meyer PRF). We refer to [4] for formal definitions of KDI-security for encryption.

Consider the "canonical" construction of symmetric encryption from PRF. Namely, given a PRF $f_s(\cdot)$ we define

$$\mathrm{Enc}_s(x) = (r, x \oplus f_s(r)) \tag{9}$$

where $r$ is chosen at random with each encryption. This encryption scheme is CPA-secure (up to the birthday bound on $|r|$) if $f_s(\cdot)$ is a secure PRF, and, intuitively, it appears that it "should" also be KDI-secure. In particular, the scheme in Eq. (9) can be shown to be KDI-secure with respect to *all functions of the key* $s$ in the ideal-cipher model or when $f$ is modeled as a family of random functions. (The proof is a simplified variant of Theorem 5.1 from [4] ,where a related construction was shown to be KDI-secure against chosen-ciphertext attacks.)

We demonstrate, however, that this construction is not KDI-secure in general in the standard model. Moreover, and perhaps more surprising it even fails for practical PRFs. Specifically, we show that when the underlying PRF is itself implemented from a block cipher via the Davies-Meyer construction, the resulting encryption scheme is not KDI-secure, even with respect to the identity function. Recall the Davies-Meyer construction

$$f_s(x) = E_x(s) \oplus s \tag{10}$$

This construction was meant as a component of a collision-resistant hash function, but for contemporary block ciphers one can expect it to also be a good PRF (in particular, this is an assumption underlying the analysis of HMAC and it holds when $E$ is modeled as an "ideal cipher"). Plugging the Davies-Meyer construction in Eq. (9) we obtain the encryption scheme $\mathrm{Enc}_s(x) = (r, x \oplus (E_r(s) \oplus s))$. An attacker that asks to encrypt the secret key will get $\mathrm{Enc}_s(s) = (r, s \oplus (E_r(s) \oplus s)) = (r, E_r(s))$, from which it can recover $s$ (using the decryption routine $E^{-1}$ with $r$ as a key). Note also that this construction fails even if $E$ is an ideal cipher!

## References


[1] IEEE P1619.* email archive. `http://grouper.ieee.org/groups/1619/email/`.

[2] B. Barak. Symmetric PRFs. Personal communications, 2001.

[3] M. Bellare and T. Kohno. A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. In *Advances in Cryptology – EUROCRYPT '03*, volume 2656 of *LNCS*, pages 491–506. Springer, 2003.

[4] J. Black, P. Rogaway, and T. Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In *Selected Areas in Cryptography*, volume 2595 of *Lecture Notes in Computer Science*, pages 62–75. Springer, 2002.

[5] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM Journal on Computing*, 13:850–864, 1984.

[6] D. Boneh and R. Ostrovsky. Circular encryption. Unpublished, 2003.

[7] R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003. Preliminary version in Crypto'98.

[8] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33(4):210–217, 1986.

[9] C. Hall, D. Wagner, J. Kelsey, and B. Schneier. Building PRFs from PRPs. In *Advances in Cryptology – CRYPTO'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 370–389. Springer, 1998.

[10] IEEE P1619. Standard for cryptographic protection of data on block-oriented storage devices. Draft standard, availabe temporarily from `http://ieee-p1619.wetpaint.com/page/IEEE+Project+1619+Home`, 2007.

[11] M. Liskov, R. L. Rivest, and D. Wagner. Tweakable block ciphers. In *Advances in Cryptology – CRYPTO '02*, volume 2442 of *Lecture Notes in Computer Science*, pages 31–46. Springer, 2002.

[12] M. Luby and C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal of Computing*, 17(2), Apr. 1988.

[13] S. Lucks. The sum of PRPs is a secure PRF. In *Advances in Cryptology - EUROCRYPT'00*, volume 1807, pages 470–484. Springer, 2000.

[14] P. Rogaway. Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. In *Advances in Cryptology - ASIACRYPT'04*, volume 3329 of *Lecture Notes in Computer Science*, pages 16–31. Springer, 2004.

# A Proof sketches

**Theorem 1** Let $g$ be any Boolean circuit *with no* $\Pi$*-gates or* $\Pi^{-1}$*-gates*. Then the construction $f_s(x) = \Pi_s(x)$ is a KDI-secure pseudorandom function in the "ideal cipher model" with respect to the singleton class $\mathcal{C} = \{g\}$.

**Proof Sketch** The attacker $\mathcal{A}$ has access to three oracles: $\Pi(\cdot,\cdot)$, and $\Pi^{-1}(\cdot,\cdot)$ that represent the ideal cipher and $f(\cdot)$ which is either $\Pi(s,\cdot)$ for a random $s$ (the "real case"), or an independent random function (the "random case"). In addition, the attacker is given the value $f(g(s))$, where $s$ is the key in the "real case" and just a random string in the "random case".

We consider a "hybrid case" which is just like the "random case", except that $f$ is chosen as a random permutation rather than a random function. Clearly, the "hybrid" and the "random" cases cannot be distinguished upto the birthday bound. The heart of the proof is in showing that the attacker cannot distinguish the "hybrid" from the "real" case.

Next we argue that the attacker has only a negligible probability to ever query its $\Pi$ or $\Pi^{-1}$ oracles with the correct key $s$: since $g$ is independent of $\Pi, \Pi^{-1}$ then all the values that the attacker sees are entries of $\Pi, \Pi^{-1}$ that by themselves are independent of $s$. (This is where the counterexample $g(s) = \Pi^{-1}(s,s)$ comes in, dependent on $\Pi^{-1}$ allows the attacker to ask for "the value in the entry in which $s$ is written".) As long as the attacker still did not query $\Pi$ or $\Pi^{-1}$ with the right key $s$, then the answer that it got so far can be completely simulated by the attacker itself, save for cases where a query $f(x)$ on some string $x$ happened to return the same value as $f(g(s))$. This last event either happens with negligible probability (if the pre-image of $g(s)$ is smaller than $2^{n/2}$) or they still leave exponentially many possibilities for $s$ (if the pre-image of $g(s)$ is larger). Hence the attacker only has an exponentially small probability of hitting the right key $s$ in the next query that it makes.

But short of querying $\Pi, \Pi^{-1}$ on the right $s$ (and since $g$ is independent of $\Pi, \Pi^{-1}$), the answers that the attacker gets in both the "hybrid" and the "real" cases are drawn from the same probability distribution. Namely the initial value of $f(g(s))$ and the answers to all the queries to $f$ are computed using a random permutation which is independent of the queries that the attacker makes to $\Pi, \Pi^{-1}$. ■

**Lemma 2** The construction $F^{(g)}$ from Eq. (1) is a KDI-secure PRF with respect to the singleton class $\mathcal{C} = \{g\}$, provided that the family $f$ is a secure PRF and has the property that for a random $s$, $g(s)$ is unpredictable even given oracle access to $f_s(\cdot)$.

**Proof Sketch** Given a KDI-attacker $\mathcal{A}^{(g)}$ against $F^{(g)}$ with respect to $\mathcal{C} = \{g\}$, we construct a distinguisher $\mathcal{B}$ against the underlying $f$. $\mathcal{B}$ uses its oracle access to $f$ to answer $\mathcal{A}^{(g)}$'s queries to $F$ and the functional query $g(s)$ to $F'$: When $\mathcal{A}^{(g)}$ queries $F(x)$ then $\mathcal{B}$ queries $f(1|x)$, and when $\mathcal{A}^{(g)}$ queries $F'(g)$ then $\mathcal{B}$ queries $f(0)$.

What happens, however, if $\mathcal{A}^{(g)}$ queries the value $x = g(s)$ from the function oracle $F$? In this case, $\mathcal{B}$, as described, responds with $f(1|x)$ while it should have answered $f(0)$. But due to the unpredictability requirement the probability that $\mathcal{A}^{(g)}$ will query $F$ on $g(s)$ is negligible. Thus, $\mathcal{B}$'s advantage against $f$ is the same as $\mathcal{A}^{(g)}$'s advantage against $F$ except for a negligible prediction probability. ■