J. Cryptology (2002) OF1–OF24
DOI: 10.1007/s00145-002-0007-1

Journal of
CRYPTOLOGY

Efficient Amplification of the Security of Weak Pseudo-Random Function Generators

Steven Myers
Department of Computer Science, University of Toronto,
10 King's College Road, Toronto, Ontario, Canada M5S 3G4
myers@cs.toronto.edu

Communicated by Moni Naor

Received March 2000 and revised July 2002
Online publication 6 December 2002

Abstract. We show that given a PRFG (pseudo-random function generator) G that is $(1/n^c)$-partially secure there exists a polynomial p such that the construction $g_1(x \oplus r_1) \oplus \cdots \oplus g_{p(n)}(x \oplus r_{p(n)})$ produces a strongly secure PRFG, where $g_i \in G$ and r_i are strings of random bits, and the key for the new PRFG is composed of the r_i's and keys for the g_i's. This is the first "natural" construction of a (totally secure) PRFG from a partially secure PRFG. Using results of Luby and Rackoff, this result also demonstrates how to construct a PRPG "naturally" from a partially secure PRPG.

Key words. Pseudo-randomness, Security amplification, Function generators, XOR Lemma.

1. Introduction

Cryptographers have noted that the Data Encryption Standard (DES) is effectively the composition of 16 insecure permutation generators. Because DES has withstood much cryptanalysis it is often considered to be a secure (given its small key size) pseudo-random permutation generator (PRPG). Similarly, AES, the predecessor of DES, is also based on the composition of at least nine insecure permutation generators, and, to date, it too has withstood much cryptanalysis. These constructions have led some cryptographers to attempt to provide evidence that supports the apparent observation that the composition of permutation generators can amplify security.

Following this line of research, Luby and Rackoff [12] defined the notion of a partially secure PRPG to be a permutation generator that produces permutations that cannot be efficiently distinguished from random permutations by small circuits with a probability better than $1/c$, for some constant $c > 1$. They proved that the composition of a constant number of partially secure PRPGs results in a partially secure PRPG with stronger security than any of its constituent components. Unfortunately, Luby and Rackoff's result did not permit the construction of a PRPG from a partially secure PRPG.

It was known that the existence of a partially secure PRFG implied a totally secure PRFG. The construction used the following chain of results. It is possible to construct a weak one-way function from a partially secure PRFG; then, using Yao's lemma [17], [11], construct a one-way function; then, using the HILL result [7], construct a pseudo-random number generator (PRNG); then, using GGM [5], construct a PRFG. However, this construction is obviously neither "natural" nor efficient.

In this paper we give a natural, efficient and parallelizable construction for generating a PRFG from a partially secure PRFG. Our proof follows from the ideas of Luby and Rackoff [12], [2]. Further, since partially secure PRPGs are a special case of partially secure PRFGs, we can use a partially secure PRPG to construct a PRFG. Then, using a previous result by Luby and Rackoff [13], or more recent work by Naor and Reingold [16], we can "naturally" and efficiently construct a PRPG from the PRFG. If $\mathsf{F} = \{\mathsf{F}^n \mid n \in \mathbb{N}\}$ is a "partially secure" PRFG, then our construction is as follows:

$$f_1^n(x \oplus r_1) \oplus \cdots \oplus f_m^n(x \oplus r_m),$$

where the f_i^n's are randomly chosen from F^n, and the r_i's are randomly chosen from $\{0,1\}^n$. The key for this new generator consists of all the keys for the functions (f_i's), and all of the strings of random bits (r_i's).

Our construction is similar to an XOR product, and in this light our proof might be considered an XOR lemma for a PRFG. Further support for this this view is found in the fact that our proof closely follows those of Levin's [11] and Luby and Rackoff's [12], where Luby and Rackoff already followed the proof of Levin.

Given that there are relatively few proofs that show security amplification in an unrestricted adversarial model, we think this result will be of interest to those researchers interested in security amplification. Further, we believe that this result can be viewed as one step in the long journey to developing a good theory for the development of block-ciphers. Currently, block-ciphers are developed primarily using heuristics, with little theory to guide the development of their underlying architecture. However, there are pragmatic examples of how the proposed construction might be of use to block-cipher designers. For instance, one pragmatic example of a partially secure PRFG might be block-ciphers with a large set of weak keys which make the cipher easy to break in some form or another. A block-cipher designer could use the construction on the cipher and reasonably expect to reduce the fraction of its weak keys. In practice there are examples of ciphers which have large numbers of weak keys. For instance, the block-cipher IDEA has 2^{51} weak keys [3]. Of course, as the total number of keys is 2^{128} this is not problematic in this example because a randomly chosen key is very unlikely to be weak.

1.1. *Related Work*

There are very few results in cryptography that demonstrate the amplification of security in a general, nonrestrictive adversarial model. The first such result was Yao's XOR Lemma [17], which now has several proofs [11], [8], [6]. All of these results apply to the security amplification of weak one-way functions and weakly unpredictable predicates. In a domain closer to that of a PRFG, Luby and Rackoff [12] give a direct product lemma for a PRPG, where the direct product is taken via the composition of a weak PRPG. Unfortunately, their proof falls short of demonstrating that the direct product of

a sufficient number of weak PRPGs yields a strongly secure PRPG, and the reasons for this case are explained in further detail in Section 3. Another direct product theorem for a PRFG is given by Myers [15], where the direct product is based on the composition and exclusive-or of a PRFG. Unfortunately, this result also fails to achieve a strongly secure PRFG for reasons similar to those of Luby and Rackoff [12], [2]. Further complicating matters with Myers' [15] result is the fact that the size of the constructed generator is super-polynomial after $\omega(\log n)$ applications of the direct product, and this further restricts the amount of security amplification that can be performed. Therefore, our result presents the first efficient and natural direct product theorem achieving a strongly secure PRFG from a weakly secure PRFG in a general adversarial model.

Since Luby and Rackoff proposed their partial security model in [12], cryptographers have developed other models where it is possible to demonstrate some forms of security amplification. Kilian and Rogaway [10] propose a model where component permutation generators are replaced with completely random permutation generators. Constructions using the generators are then analyzed, and their security compared with that of a *random* permutation generator. Note that in this model, since the permutation generators are random, attacks can only be performed on the construction, and not on the underlying component generators. Kilian and Rogaway call such attacks *generic*, as they do not make use of the underlying structure of the permutation generator. In this model Kilian and Rogaway [10] show that the DESX construction increases the effective key length of DES. Also in the same model, Aiello et al. [1] have shown that the composition of multiple random permutation generators results in a permutation generator that is more secure against generic attacks than an individual constituent permutation generator.

2. Notation, Definitions and the Model

Below we introduce some notation and terminology that are used in the paper.

Notation 1. For $\mu, \nu \in \{0,1\}^*$, let $\mu \bullet \nu$ denote their concatenation.

Notation 2. Let $\mathcal{F}^{l \to p}$ denote the set of all functions $f\colon \{0,1\}^l \to \{0,1\}^p$, and let $\mathcal{F}^n$ be the set $\mathcal{F}^{n \to n}$.

Notation 3. For $\alpha, \beta \in \{0,1\}^n$, let $\alpha \oplus \beta$ denote the bit-by-bit exclusive-or of α and β. For $f, g \in \mathcal{F}^n$, let $(f \oplus g)(\alpha)$ denote $f(\alpha) \oplus g(\alpha)$.

Notation 4. Let $f \in \mathcal{F}^{n \to m}$ and $r \in \{0,1\}^n$. Then define the function $f^r_{\oplus}\colon \{0,1\}^n \to \{0,1\}^m$ as $f^r_{\oplus}(x) = f(x \oplus r)$ for all $x \in \{0,1\}^n$.

Notation 5. For any set A, let $x \in A$ be the action of uniformly at random choosing an element x from A. For any distribution $\mathcal{D}$, let $x \in \mathcal{D}$ be the action of randomly choosing an element according to $\mathcal{D}$. It will be clear from context when $\in$ is used to refer to an element in a set, and when it refers to choosing from a distribution.

Definition 1. Let $\mathcal{D}_1, \mathcal{D}_2, \ldots$ be a sequence of distributions, and let e represent a series of events $e_1, e_2, \ldots$ such that for all i, e_i is an event of D_i. We say that e occurs with **significant probability** if for some constant $c > 0$ and for infinitely many n the $\Pr_{\mathcal{D}_n}(e_n) \geq 1/n^c$. We say that an event e occurs with **negligible probability** if, for all constants $c > 0$ and for all sufficiently large n, $\Pr_{\mathcal{D}_n}(e_n) < 1/n^c$.

2.1. *Circuits*

In the definition of each cryptographic primitive there exists the notion of an adversary. Abstractly, its purpose is to break an effect that a primitive is trying to achieve. Resource bounds are imposed on the adversaries, so that they model the computational power "real world" adversaries might feasibly have access to. There are two standard computational models that are used to define resource bounded adversaries: uniform and non-uniform. To clarify the proof and make it more concise we present our results with respect to non-uniform adversaries. In Section 4.1 we discuss some of the more pertinent changes required in order to make the proof go through in the uniform model.

A non-uniform adversary is a sequence of circuits $(C_1, C_2, \ldots)$, where circuit C_i is used on inputs of size i. We wish to model efficient computation on the part of the adversary, so we assume that the size of each circuit C_i is bounded by $p(i)$, for some polynomial p. The size of a circuit is defined to be the number of gates and the number of connections between gates in the circuit. For simplicity we assume we have gates for all 16 binary and 4 unary functions.

In order to model the adversaries of certain primitives we allow the circuits to have access to an oracle. This is modeled by defining oracle gates to be gates of unit size that compute a specified function. The gates are otherwise treated like normal gates. An oracle function will normally be considered an input to the circuit.

We stress that the description of the circuit family need not be efficiently computable, even though each circuit is of small size relative to the size of its input.

Definition 2. Let C be a circuit whose outputs are in the range $\{0, 1\}$. Then we say C is a **decision circuit**. Let x be an input to C. Then we say C **accepts** x if $C(x) = 1$, and we say that C **rejects** x if $C(x) = 0$.

Definition 3. We say a circuit C is **probabilistic** if it requires as input a sequence of random bits in order to allow it to make random choices during its computation.

Notation 6. Let $\mathcal{D}$ be a distribution over the inputs of a decision circuit C. Then we use as a shorthand $\Pr_{d \in \mathcal{D}}(C(d))$ to represent $\Pr_{d \in \mathcal{D}}[C(d) = 1]$.

Definition 4. Let $\mathcal{D}$ be a distribution over the inputs of a decision circuit C. We say that C **accepts** a **fraction** $\Pr_{d \in \mathcal{D}}(C(d))$ of its inputs, and **rejects** a **fraction** $1 - \Pr_{d \in \mathcal{D}}(C(d))$ of its inputs.

Notation 7. We write C^f to represent a circuit C that has oracle gates that compute the function f in unit time. We wish to consider this function as an "input" to the circuit,

and therefore if f is of the form $\{0,1\}^n \to \{0,1\}^{m(n)}$, for a polynomial m, then we say that f is part of C's input and it has size n.

Notation 8. Let C be a circuit with access to the oracle function f. Then let Q_C denote the number of oracle gates in C. (Note: Q is short for query.)

In the remainder of the paper we assume that all circuits are standardized in the following manner: no circuit will repeat oracle queries, and all circuits C_n in a circuit family $\{C_n\}$ will perform exactly $m(n)$ queries, for some polynomial m (that is $Q_{C_n} = m(n)$). Any polynomial-sized family of circuits can be easily modified to satisfy the above two requirements.

2.2. *Function Generators*

Definition 5. We call $G\colon \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^m$ a function generator. We say that $k \in \{0,1\}^\kappa$ is a **key** of G, write $G(k,\cdot)$ as $g_k(\cdot)$ and say that key k chooses the function g_k. Let $g \in G$ represent the act of uniformly at random choosing a key k from $\{0,1\}^\kappa$, and then using the key k to choose the function g_k.

Let m and ℓ be polynomials, and let $\mathcal{N} \subseteq \mathbb{N}$ be an infinitely large set. For each $n \in \mathcal{N}$, let $G^n\colon \{0,1\}^{\ell(n)} \times \{0,1\}^n \to \{0,1\}^{m(n)}$ be a function generator. We call $G = \{G^n \mid n \in \mathcal{N}\}$ a function generator ensemble.

In an abuse of notation, we refer to both specific function generators and ensembles as function generators.

Definition 6 (ε-Distinguishing Adversary). Let $\varepsilon\colon \mathbb{N} \to [0,1]$, and let $\mathcal{D}_1 = \{\mathcal{D}_1^i \mid i \in \mathbb{N}^+\}$ and $\mathcal{D}_2 = \{\mathcal{D}_2^i \mid i \in \mathbb{N}^+\}$ be two sequence of distributions over oracle gates, where $\mathcal{D}_j^i$ is a distribution over oracle gates of input size i for $j \in \{1,2\}$. If $\{C_n\}$ is an adversary with access to oracle gates, then we say it is capable of ε**-distinguishing** $\mathcal{D}_1$ **from** $\mathcal{D}_2$ if, for some polynomial p and infinitely many n,

$$\left| \Pr_{d_1 \in \mathcal{D}_1}[C_n^{d_1} = 1] - \Pr_{d_2 \in \mathcal{D}_2}[C_n^{d_2} = 1] \right| \geq \varepsilon(n) + \frac{1}{p(n)}.$$

Notation 9. If a circuit family $\{C_n\}$ ε-distinguishes $\mathsf{G} = \{\mathsf{G}^n \mid n \in \mathbb{N}\}$ from $\{\mathcal{F}^n \mid n \in \mathbb{N}\}$, then we say $\{C_n\}$ ε**-distinguishes** G.

Definition 7 (Pseudo-Random Function Generator Ensembles). Let m and ℓ be polynomials. For each n let $\mathsf{G}^n\colon \{0,1\}^{\ell(n)} \times \{0,1\}^n \to \{0,1\}^{m(n)}$ be a **function generator**, computable in time bounded by a polynomial in n. Define $\mathsf{G} = \{\mathsf{G}^n \mid n \in \mathbb{N}\}$ to be the function generator ensemble. Define $\mathcal{F} = \{\mathcal{F}^{n \to m(n)} \mid n \in \mathbb{N}\}$. We say that G is $(1 - \varepsilon(n))$ **secure** if there exists no adversary $\{C_n\}$, bound in size to be polynomial in n, which can ε-distinguish G from $\mathcal{F}$. We say that G is a **pseudo-random function generator** (PRFG) if it is 1 secure.

Definition 8. If G is a 1-secure generator, we say it is **strongly secure**. If G is $1/p(n)$ secure, for some polynomial p, then we say that it is **partially secure**. If G is not partially secure, then we say it is **insecure**.

2.3. *The Chernoff Bound*

Below is a well known form of the Chernoff Bound. For a proof of this result refer to [14] or any standard book on probabilistic computation.

Lemma 1 (Chernoff Bound). *Let $x_1, x_2, x_3, \ldots$ be identical independently distributed random variables that take the values 0 or 1 with probabilities q or $p = 1 - q$, respectively. Let $X_{n^t} = (1/n^t) \sum_{i=1}^{n^t} x_i$. Then for any k and l, there exists a t such that*

$$\Pr\left[|X_{n^t} - p| \geq \frac{1}{n^k}\right] \leq \frac{1}{2^{n^l}}.$$

3. Result

We show that there is a "natural" construction that builds strongly secure PRFGs from partially secure PRFGs. The construction we present uses function generators that generate functions of the form $f\colon \{0,1\}^n \to \{0,1\}^n$, this is done to simplify the presentation, and the result can be easily modified to generate functions of the form $f\colon \{0,1\}^n \to \{0,1\}^{m(n)}$, for any polynomial m. The construction is based on the operator generator described below.

Let f_1 and f_2 be two functions of the form $\{0,1\}^n \to \{0,1\}^n$. For each $r_1, r_2 \in \{0,1\}^n$ define the operator $\diamond^n_{r_1 \bullet r_2}$, which acts on the functions f_1 and f_2 and produces a function of the form $\{0,1\}^n \to \{0,1\}^n$ as defined below:

$$(f_1 \diamond^n_{r_1 \bullet r_2} f_2)(x) = f_1(x \oplus r_1) \oplus f_2(x \oplus r_2).$$

We define the $\diamond$ operator generator (read Diamond) as $\diamond = \{\diamond^n_{r_1, r_2} \mid n \in \mathbb{N} \wedge r_1, r_2 \in \{0,1\}^n\}$.

Before describing the construction we formally describe how to combine two function generators using the $\diamond$ operator generator.

Definition 9. Let $\mathsf{G} = \{\mathsf{G}^n\colon \{0,1\}^{\ell(n)} \times \{0,1\}^n \to \{0,1\}^n \mid n \in \mathbb{N}\}$ and $\mathsf{H} = \{\mathsf{H}^n\colon \{0,1\}^{\kappa(n)} \times \{0,1\}^n \to \{0,1\}^n \mid n \in \mathbb{N}\}$ be function generator ensembles. Let $\diamond$ be the operator generator defined previously. Let $\mathsf{F} = \{\mathsf{F}^n\colon \{0,1\}^{\ell(n)+\kappa(n)+2\cdot n} \times \{0,1\}^n \to \{0,1\}^n \mid n \in \mathbb{N}\}$ be the function generator defined by $\mathsf{F}^n(k_1 \bullet k_2 \bullet k_3 \bullet k_4, x) = (g^n_{k_1} \diamond^n_{k_3 \bullet k_4} h^n_{k_2})(x)$, where $|k_1| = \ell(n)$, $|k_2| = \kappa(n)$ and $|k_3| = |k_4| = n$. This is written in shorthand as $\mathsf{F} = \mathsf{G} \diamond \mathsf{H}$.

Similarly, if $g\colon \{0,1\}^n \to \{0,1\}^n$, then we write $g \diamond \mathsf{H}$ as short-hand for the function generator defined by $\mathsf{F}^n(k_2 \bullet k_3 \bullet k_4, x) = (g \diamond^n_{k_3 \bullet k_4} h^n_{k_2})(x)$, where $|k_2| = \kappa(n)$ and $|k_3| = |k_4| = n$.

3.1. *The Construction*

Let p be a polynomial. We construct the generator F from the generator G as follows:

$$\mathsf{F} = \underbrace{\mathsf{G}\diamond\cdots\diamond\mathsf{G}}_{p(n)}.$$

Note that in order to compute a random function $f \in \mathsf{F}^n$ it is sufficient to compute

$$(g_1(x \oplus r_1) \oplus \cdots \oplus g_{p(n)}(x \oplus r_{p(n)})),$$

where $g_i \in \mathsf{G}^n$ and $r_i \in \{0,1\}^n$. Observe that the key for F includes $p(n)$ keys for G and $p(n)$ random strings.

The random strings used in the construction are necessary for the security amplification, and if they are omitted a counter-example to our security amplification claims can be easily constructed, as follows. Let F be a *secure* PRFG from which we construct a $\frac{1}{2}$-secure generator G. For each $f \in \mathsf{F}$ we have a corresponding function $g \in \mathsf{G}$ that is defined as

$$g(x) = \begin{cases} FBZ(f(x)) & \text{if } x = \bar{0}, \\ f(x) & \text{otherwise,} \end{cases}$$

where $\bar{0}$ represents a binary string of all 0's and *FBZ* is a predicate that takes a string and returns the same string with the first bit set to 0 (that is, $FBZ(x_1, \ldots, x_n) = 0, x_2, \ldots, x_n$). Intuitively this generator is $\frac{1}{2}$-secure as only half of the random functions $f \in \mathcal{F}$ will have $f(\bar{0})$ commencing with the bit 0, whereas all of the functions $g \in \mathsf{G}$ will have this property. Excluding this distinction G is a duplicate of F, and therefore its functions are indistinguishable from random functions, and so this is the only advantage a distinguisher has in distinguishing G from $\mathcal{F}$.

Consider the construction $\mathsf{H} = \mathsf{G} \oplus \mathsf{G}$, this is the same construction as $\mathsf{G}\diamond\mathsf{G}$, but *without* the random offsets r_1 and r_2. It has the property that for every $h \in \mathsf{H}$ there exists $g_1, g_2 \in \mathsf{G}$ such that $h = g_1 \oplus g_2$, and therefore $h(\bar{0})$'s first bit will always be 0: ensuring that H is not more than $\frac{1}{2}$-secure. This demonstrates that the offsets in the Diamond construction are in fact necessary for security amplification. We stress that our construction requires that both offsets be chosen randomly and thus are most likely different.

For further discussion on this construction and several other plausible candidates for security amplification see [15].

3.2. *Presentation*

We proceed to proving the security amplification properties of the above construction. First we discuss some issues relating to how such proofs will be presented.

The arguments will be presented in a top-down fashion. We will begin by giving the intuition for the highest level of an argument, and then provide the technical details. Required lemmas will be stated, but in many cases proofs of the lemmas will be delayed until a later point in the paper.

In what follows, there are many proofs that deal with values that are approximations to the values of interest. Because of this, in many of the calculations there are many terms

involving inverse polynomials that represent possible approximation errors. While it is important to ensure the calculations are correct with these error bounds, they are cumbersome and make the calculations seem more complicated than the underlying intuition suggests. Therefore, when intuitive explanations are presented we ignore all of the inverse polynomial approximation errors, and assume we are only dealing with the exact values we are trying to approximate.

We will be presenting asymptotic arguments that hold for either infinitely many or all sufficiently large n. For clarity, in all of the formal statements of claims, lemmas and theorems all appropriate indices will be included, but the index n will be fixed, where appropriate, in the proofs of such statements in order to simplify notation. Therefore, the following will hold when the index n is fixed: for any function $\varepsilon\colon \mathbb{N} \to \mathcal{R}$, from the Naturals to a given range $\mathcal{R}$, we use the ε to represent $\varepsilon(n)$; for an arbitrary family of circuits $\{C_n\}$ we denote C_n by C; and for an ensemble of function generators $\{\mathsf{G}_n\}$ we denote G_n by G. Finally, when presenting the intuition for a given argument we will be far more relaxed with the index n, often dropping it completely to prevent notation from obscuring the underlying intuition. It should be clear to the reader where the appropriate indices would appear.

3.3. *Results*

In order to prove the security of the construction we use the Diamond Isolation Lemma (the name for this lemma comes from the stylistically similar Isolation Lemma used by Levin [11] in proving Yao's XOR Lemma [17]). Intuitively, the lemma shows that the function generator that results from the combination of two partially secure function generators by the $\diamond$ operator generator is more secure than either of the two constituent generators. More specifically, if there are two generators, G and H, that respectively cannot be ε- and δ-distinguished, then the generator $\mathsf{G}\diamond\mathsf{H}$ cannot be $\varepsilon\delta$-distinguished. As we will apply the Isolation Lemma iteratively, it is easier to work with if the lemma is stated in the contrapositive. Also, because of the iterative application of the lemma we need to be concerned about the sizes of the distinguishing circuits for G and H. This is the reason for the gross asymmetry in the size of the circuits that appears in the statement of the lemma. The need for the differences in circuit size will be made clear in the proof of Theorem 1. The majority of the work in this paper goes towards proving the lemma correct.

Lemma 2 (Diamond Isolation Lemma). *There exists a fixed polynomial p (that is retrievable from the proof) such that the following hold. Let $\varepsilon, \delta\colon \mathbb{N} \to [0, 1]$ be functions. Let there exist a constant $d > 0$ such that for all sufficiently large n, $\varepsilon(n) \leq 1 - 1/n^d$. Let H and G be function generators, where $c_{\mathsf{G}}(n)$ and $c_{\mathsf{H}}(n)$ are polynomials which bound from above the size of the circuits which compute the function generators, respectively.*

Hypothesis: *Let s_C be a polynomial. There exists a family of decision-circuits $\{C_n\}$, where for each n the circuit C_n is of size bounded above by $s_C(n)$, and there exists $c > 2d$ such that for infinitely many n,*

$$\left| \Pr_{g \in \mathsf{G}^n \diamond \mathsf{H}^n}(C_n^g) - \Pr_{f \in \mathcal{F}^n}(C_n^f) \right| \geq \varepsilon(n)\delta(n) + \frac{1}{n^c}.$$

Conclusion: *For infinitely many n there exists either a decision-circuit* Υ_n *of size* $p(n^c \cdot c_{\mathsf{G}}(n))s_C(n)$ *for which*

$$\left| \Pr_{h \in \mathsf{H}^n}(\Upsilon_n^h) - \Pr_{f \in \mathcal{F}^n}(\Upsilon_n^f) \right| \geq \varepsilon(n) + \frac{1}{n^{3c}},$$

or a decision-circuit Ξ_n *of size* $\leq (2Q_{C_n}c_{\mathsf{H}}(n) + s_C(n))$, *where* $Q_{\Xi_n} = Q_{C_n}$, *and for which*

$$\left| \Pr_{g \in \mathsf{G}^n}(\Xi_n^g) - \Pr_{f \in \mathcal{F}^n}(\Xi_n^f) \right| \geq \delta(n) + \frac{1}{n^c}.$$

(*We remind the reader that* Q_C *represents the number of queries performed by circuit* C.)

Akcoglu, Luby and Rackoff prove a similar lemma in [12] and [2]. It shows that the composition of two partially secure PRPGs results in a generator that is more secure than either of its constituents. Excluding the fact that their lemma is restricted to permutation generators instead of function generators and that they are concerned with composition as opposed to the $\diamond$ operator, our lemma is stronger in two senses. First, the security requirement in the hypothesis is strictly weaker (i.e. the improvement in security from combining the two generators is stronger in our result). Second, the size of the distinguishing circuit for G is only additively larger than the distinguishing circuit for $\mathsf{G}\diamond\mathsf{H}$. In the Luby and Rackoff construction the distinguishing circuits for G and H are both multiplicatively larger than the circuit that distinguishes $\mathsf{G} \circ \mathsf{H}$. It is this second fact that permits our construction to achieve PRFGs, but prevents Luby and Rackoff from achieving more than a constant number of security amplifying compositions. Furthermore, our proof is simpler than that of Luby and Rackoff. This is due to the fact that their proof contains a corollary that corresponds to Lemma 7 in our proof, but, unlike Lemma 7, their corollary is only proven true with respect to the computational security of $\mathsf{G} \circ \mathsf{H}$. This restriction is necessary for their construction, but increases the difficulty of the proof. We now prove that our construction produces a PRFG from a $1 - \varepsilon$ secure PRFG.

Theorem 1 (Diamond Composition Theorem). *Let* $\varepsilon\colon \mathbb{N} \to (0, 1)$ *be a function, where* ε *is bounded away from* 1 *by an inverse polynomial. Let* G *be a polynomial. Let* G *be a* $1 - \varepsilon(n)$ *secure PRFG. The generator* $\mathsf{F} = \underbrace{\mathsf{G}\diamond\cdots\diamond\mathsf{G}}_{p(n)}$ *is a* $1 - \varepsilon(n)^{p(n)}$ *secure PRFG.*

Proof. The intuition for this argument is as follows. Assume that F does not have the claimed security, and thus there is a family of distinguishing circuits for F that breaks the claimed security. We apply the Isolation Lemma to the generator F. The result is either that the generator G is not $1 - \varepsilon$ secure as claimed, or we have a family of distinguishing circuits (only slightly larger than the original circuit family) for a generator that is constructed from fewer applications of the $\diamond$ operator (and thus fewer constituent G generators). We apply the Isolation Lemma inductively to this smaller generator until we are left with an $\varepsilon(n) + 1/n^c$ family of distinguishing circuits for the remaining generator G, which contradicts its assumed $1 - \varepsilon(n)$ security.

For sufficiently large n and some $d > 0$ let $\varepsilon(n) < 1 - 1/n^d$. Assume for contradiction that there exists a distinguishing family of circuits $\{D_n\}$ and a $c > 2d$ such that for infinitely many n, $|\Pr_{f\in\mathsf{F}^n}(D_n^f) - \Pr_{f\in\mathcal{F}^n}(D_n^f)| \geq \varepsilon(n)^{p(n)} + 1/n^c$.

For each i define the generator $\mathsf{F}_i = \underbrace{\mathsf{G}\diamond\cdots\diamond\mathsf{G}}_{i}$. For $1 \leq i \leq p(n)$, define the predicate $\mathcal{P}(i)$ as: there exists a family of circuits $\{D_{i,n}\}$ such that for infinitely many n,

$$\left|\Pr_{f\in\mathsf{F}_i^n}(D_{i,n}^f) - \Pr_{f\in\mathcal{F}^n}(D_{i,n}^f)\right| \geq \varepsilon(n)^i + \frac{1}{n^c},$$

where each circuit in the family is of size $(2Q_{D_n}\cdot c_{\mathsf{G}}(n))(p(n)-i)+s_D(n)$ and performs Q_{D_n} oracle queries.

We prove $\mathcal{P}(i)$ true for $1 \leq i \leq p(n)$ by (reverse) induction.

Base Case ($\mathcal{P}(p(n))$). By definition $\mathsf{F} = \mathsf{F}_n$ and by defining $\{D_{p(n),n}\} = \{D_n\}$, we have $|\Pr_{f\in\mathsf{F}^n_{p(n)}}(D^f_{p(n),n}) - \Pr_{f\in\mathcal{F}^n}(D^f_{p(n),n})| \geq \varepsilon(n)^{p(n)} + 1/n^c$. Further, the restrictions on the size and number of queries performed by the circuits in $\{D_{p(n),n}\}$ are easily seen to hold, proving $\mathcal{P}(p(n))$.

Inductive Step. We assume the inductive hypothesis, $\mathcal{P}(i)$, holds for a particular $1 < i \leq p(n)$, and prove it true for $i-1$. By the inductive hypothesis there exists a family of $\varepsilon(n)^i$-distinguishing circuits that have size $(2Q_{D_n}\cdot c_{\mathsf{G}}(n))(p(n)-i)+s_D(n)$ and perform Q_{D_n} oracle queries.

We observe that $\mathsf{F}_i = \mathsf{F}_{i-1}\diamond\mathsf{G}$, and therefore we can apply the Isolation Lemma to F_i with the family of distinguishing circuits $\{D_{i,n}\}$. Thus, either there exists a polynomial in n sized family of circuits $\{\Upsilon_n\}$ that break the assumed $1-\varepsilon(n)$ security of G, causing a contradiction; or there exists a family of circuits $\{D_{i-1,n}\}$ such that

$$\left|\Pr_{f\in\mathsf{F}^n_{i-1}}(D^f_{i-1,n}) - \Pr_{f\in\mathcal{F}^n}(D^f_{i-1,n})\right| \geq \varepsilon(n)^{i-1} + \frac{1}{n^c}.$$

Further, each circuit $D_{i-1,n}$ performs Q_{D_n} oracle queries and has size

$$2Q_{D_n}\cdot c_{\mathsf{G}}(n)+(2Q_{D_n}\cdot c_{\mathsf{G}}(n))(p(n)-i)+s_D(n) = (2Q_{D_n}\cdot c_{\mathsf{G}}(n))(p(n)-(i-1))+s_D(n),$$

proving $\mathcal{P}(i-1)$.

By the principle of induction, $\mathcal{P}(1)$ holds, but this implies that the family of circuits $\{D_{1,n}\}$ break the $1-\varepsilon(n)$ security of $\mathsf{F}_1 = \mathsf{G}$, proving the theorem. □

Observe the proof of Theorem 1 would not go through if the Isolation Lemma constructed distinguishers were both significantly larger than the original distinguisher. In particular, if the circuits defined by the predicate $\mathcal{P}(1)$ were super-polynomial in size, then the circuit would not break the security of $\mathsf{F}_1 = \mathsf{G}$, as the security claims of G are made relative to a polynomial-sized circuit. This observation makes the need for the asymmetry of the Isolation Lemma clear.

3.4. *Proof of the Isolation Lemma*

Intuitively, the Isolation Lemma claims that by combining two partially secure PRFGs with the $\diamond$ operator, a new generator is formed that has insecurity proportional to the product of the insecurities of the two constituent generators. The lemma is stated in the contrapositive, with concern for the size of the distinguishing circuits, so that the lemma can be applied iteratively, as in Theorem 1. At this point the reader may wish to recall the statement of the Isolation Lemma.

At the highest level, the proof of the Isolation Lemma is easy to explain: assuming we have a family of $\varepsilon\delta$-distinguishing circuits, $\{C_n\}$, for the constructed generator, $\mathsf{G}\diamond\mathsf{H}$, then we show how to construct either a family of δ-distinguishing circuits for the first constituent generator H, or a family of ε-distinguishing circuits for the second constituent generator G. Again, as we wish to apply this lemma iteratively, we need to ensure that one of the circuit families we create is not much larger than the original circuit family $\{C_n\}$.

The proof proceeds as follows, we first demonstrate (Lemma 4) that either there exists a very simple algorithm that δ-distinguishes G from $\mathcal{F}$, or it is the case that for every $h \in \mathsf{H}$ the ability of the distinguishing circuits $\{C_n\}$ to distinguish between $h\diamond\mathsf{G}$ and $\mathcal{F}$ is restricted: they can do no better than δ-distinguish. The existence of the simple algorithm would prove the Isolation Lemma, so we assume it is false and use the resulting restriction on the circuits $\{C_n\}$ (that δ-distinguish between $\mathcal{F}$ and $h\diamond\mathsf{G}$) to construct a more complicated algorithm that ε-distinguishes H from $\mathcal{F}$.

The second algorithm is developed by observing that, by assumption, the circuits in $\{C_n\}$ accept a fraction of $\mathsf{G}\diamond\mathsf{H}$ that is "significantly larger" than $\varepsilon\delta + \Pr_{f\in\mathcal{F}}(C^f)$. However, the restriction implies that for each $h \in \mathsf{H}$ not much more than a $\delta + \Pr_{f\in\mathcal{F}}(C^f)$ fraction of the functions in $\mathsf{G}\diamond h$ are accepted by C. As $\Pr_{\varphi\in\mathsf{G}\diamond\mathsf{H}}(C^\varphi)$ is the expected value of $\Pr_{\varphi\in\mathsf{G}\diamond h}(C^\varphi)$ over the distribution H, it must be the case that $\Pr_{\varphi\in\mathsf{G}\diamond h}(C^\varphi)$ is "significantly larger" than $\Pr_{f\in\mathcal{F}}(C^f)$ for at least an ε fraction of the $h \in \mathsf{H}$.

Given a function ω, our distinguishing circuit will approximate $\Pr_{\psi\in\mathsf{G}\diamond\omega}(C^\psi)$ and accept if it is "significantly larger" than $\Pr_{f\in\mathcal{F}^n}(C^f)$. By the above argument this circuit will accept an ε fraction of the functions in H. In order for the circuit to ε-distinguish functions in H from random functions, the same circuit needs to accept almost no random functions. Lemma 5 shows that almost no random functions will be accepted by this circuit, proving the Isolation Lemma. The proof of Lemma 5 is involved and therefore is not presented until after the proof of the Isolation Lemma.

The Simple Algorithm

In this section we present Lemma 4. Its proof contains the simpler algorithm used to δ-distinguish G in the Isolation Lemma. The intuition is as follows: if there exists an infinite number of $h \in \mathsf{H}$ such that C can δ-distinguish between $h\diamond\mathsf{G}$ and $h\diamond\mathcal{F}$, then we simply hard-wire the appropriate h's into the circuit. When the circuit is given a function φ, that is either in G or $\mathcal{F}$, it simulates the execution of C with the oracle $h\diamond\varphi$. Alternatively, if there is not an infinite number of such $h \in \mathsf{H}$, then C can do no better than δ-distinguish $h\diamond\mathsf{H}$ and $h\diamond\mathcal{F}$. By observing that for every $h \in \mathcal{F}$ the distributions $h\diamond\mathcal{F}$ and $\mathcal{F}$ are the same, we note that C cannot δ-distinguish $h\diamond\mathsf{H}$ from $\mathcal{F}$.

We first demonstrate that if you form a distribution on functions by combining any particular function with the uniform distribution over $\mathcal{F}$ using the $\diamond$ operator, then the resulting distribution is identical to the uniform distribution over $\mathcal{F}$.

Lemma 3. *Fix n. For each $h \in \mathcal{F}^n$, the distribution $h \diamond \mathcal{F}^n$ and the uniform distribution over $\mathcal{F}^n$ are identical.*

Proof. Fix $r_1, r_2 \in \{0,1\}^n$ and fix $h \in \mathcal{F}^n$, then for every $g \in \mathcal{F}^n$, $\Pr_{f \in \mathcal{F}^n}[h_{\oplus}^{r_1} \oplus f_{\oplus}^{r_2} = g] = \Pr_{f \in \mathcal{F}^n}[f = g]$. □

Lemma 4. *Either there exists a family of decision-circuits $\{\Xi_n\}$, where for each n the circuit Ξ_n is of size $\leq Q_{C_n} 2c_{\mathsf{H}}(n) + s_C(n)$; $Q_{\Xi_n} = Q_{C_n}$; and for infinitely many n,*

$$\left| \Pr_{g \in \mathsf{G}^n}(\Xi_n^g) - \Pr_{f \in \mathcal{F}^n}(\Xi_n^f) \right| \geq \delta(n) + \frac{1}{n^c};$$

or for all sufficiently large n and all $h \in \mathsf{H}^n$,

$$\left| \Pr_{g \in \mathsf{G}^n \diamond h}(C_n^g) - \Pr_{f \in \mathcal{F}^n}(C_n^f) \right| < \delta(n) + \frac{1}{n^c}.$$

(*We remind the reader that Q_C represents the number of oracle queries performed by circuit C.*)

Proof. Suppose it is the case that for infinitely many n there exists an $h \in \mathsf{H}^n$ such that $|\Pr_{g \in \mathsf{G} \diamond h}(C_n^g) - \Pr_{f \in \mathcal{F}^n}(C_n^f)| \geq \delta(n) + 1/n^c$. For each such n we create a decision circuit Ξ_n, where $\Xi_n^w = C_n^{(w \diamond h)}$. We observe that

$$\begin{aligned}
\left| \Pr_{\psi \in \mathsf{G}^n}(\Xi_n^\psi) - \Pr_{f \in \mathcal{F}^n}(\Xi_n^f) \right| &= \left| \Pr_{\psi \in \mathsf{G}^n \diamond h}(C_n^\psi) - \Pr_{f \in \mathcal{F}^n \diamond h}(C_n^f) \right| \\
&= \left| \Pr_{\psi \in \mathsf{G}^n \diamond h}(C_n^\psi) - \Pr_{f \in \mathcal{F}^n}(C_n^f) \right| \qquad \text{(Lemma 3)} \\
&\geq \delta(n) + \frac{1}{n^c}.
\end{aligned}$$

It is easy to see that C_n can be modified, in a straightforward manner, by adding $Q_{C_n}(C_{\mathsf{H}}(n) + 10n)$ gates and wires to compute Ξ_n, while still using Q_{C_n} oracle gates. For simplicity of presentation in this paper we have assumed that $10n \leq C_{\mathsf{H}}(n)$, giving us a circuit of size $\leq s_C(n) + Q_{C_n}(2C_{\mathsf{H}}(n))$. □

The Complicated Algorithm and the Main Argument

We now present the argument for the Isolation Lemma. It contains the more complicated distinguishing algorithm that was previously described.

We begin by assuming the hypothesis of the Isolation Lemma. Further, we drop the absolute value from the hypothesis and simply assume that for infinitely many n,

$$\Pr_{g\in \mathsf{G}^n \diamond \mathsf{H}^n}(C_n^g) - \Pr_{f\in \mathcal{F}^n}(C_n^f) \geq \varepsilon(n)\delta(n) + \frac{1}{n^c}, \tag{1}$$

for if this is not the case we can flip the output bit of C_n. By Lemma 4 (stated above), we assume that for all sufficiently large n and all $h \in \mathsf{H}^n$,

$$\left| \Pr_{\psi\in \mathsf{G}^n \diamond h}(C_n^\psi) - \Pr_{f\in \mathcal{F}^n}(C_n^f) \right| < \delta(n) + \frac{1}{n^c}. \tag{2}$$

We outline the argument and remind the reader that in such outlines we drop the index n as well as additive inverse polynomial terms. As stated earlier, we will construct a circuit that on input w approximates the value of $\Pr_{f\in \mathsf{G}\diamond w}(C^f)$ and accepts if it is sufficiently larger than $\Pr_{f\in\mathcal{F}}(C^f)$. By Lemma 5 almost no random functions will be accepted by such a circuit. It needs to be demonstrated that at least an ε-fraction of the $h \in \mathsf{H}$ will be accepted. This is done by an averaging argument. Suppose less than an ε-fraction of the $h \in \mathsf{H}$ are accepted by our circuit. For those h's that are accepted it is the case that $\Pr_{f\in \mathsf{G}\diamond h}(C^f) < \delta + \Pr_{f\in\mathcal{F}}(C^f)$, by (2). For the remainder of the h's, $\Pr_{f\in \mathsf{G}\diamond h}(C^f) < \Pr_{f\in\mathcal{F}}(C^f)$, by the fact that they are not accepted by the circuit. However, this implies that $\Pr_{f\in \mathsf{G}\diamond \mathsf{H}}(C^f) < \varepsilon(\delta + \Pr_{f\in\mathcal{F}}(C^f)) + (1-\varepsilon)\Pr_{f\in\mathcal{F}}(C^f) = \varepsilon\delta + \Pr_{f\in\mathcal{F}}(C^f)$, and this contradicts (1), the assumed security of $\mathsf{G}\diamond\mathsf{H}$.

Lemma 5. *For all families of polynomial-sized decision circuit families $\{C_n\}$, for $i > 0$ and for each n let*

$$K_n(i) = \Pr_{f\in\mathcal{F}^n}(C_n^f) + \frac{1}{n^i} \quad \textit{and let} \quad S^n(i) = \left\{ w \in \mathcal{F}^n \;\middle|\; \Pr_{g\in \mathsf{G}^n\diamond w}(C_n^g) \geq K_n(i) \right\}.$$

Then for all i,j, $\Pr_{w\in\mathcal{F}^n}(w \in S^n(i)) \leq 1/n^j$, for sufficiently large n.

The proof of Lemma 5 is presented in Section 3.5. In the remainder of this proof we assume n is fixed and sufficiently large for all inequalities to hold. Therefore, n will be dropped from notation wherever possible. We refer the reader back to the end of Section 3.2 for a reminder of how notation is modified.

We commence construction of the ε-distinguishing circuit by noting that, although for a given $w \in \mathcal{F}$ we cannot compute $\Pr_{\varphi\in \mathsf{G}\diamond\omega}(C^\varphi)$ in polynomial time, we can approximate it with the probabilistic circuit A:

$$A^w = \frac{1}{n^b}\sum_{i=1}^{n^b} C^{(g_i \diamond^n_{k_i^1 \bullet k_i^2} w)},$$

where $g_1, \ldots, g_{n^b} \in \mathsf{G}^n$ and $k_1^1, k_1^2, \ldots, k_{n^b}^1, k_{n^b}^2 \in \{0,1\}^n$ are randomly chosen. Let $\kappa(n)$ be the length of the key of H^n, and set (with foresight) $\alpha > 1$ so that $n^\alpha > \kappa(n)$. Using

the Chernoff Bound, b is chosen large enough so that

$$\Pr_{w\in\mathcal{F}^n}\left[\left|A^w - \Pr_{\varphi\in\mathsf{G}\diamond w}(C^\varphi)\right| \geq \frac{1}{n^{4c}}\right] \leq \frac{1}{2^{n^\alpha}}$$

and

$$\Pr_{h\in\mathsf{H}}\left[\left|A^h - \Pr_{\varphi\in\mathsf{G}\diamond h}(C^\varphi)\right| \geq \frac{1}{n^{4c}}\right] \leq \frac{1}{2^{n^\alpha}}.$$

Note that for all of the $h \in \mathsf{H}$,

$$\left|A^h - \Pr_{\varphi\in\mathsf{G}\diamond h}(C^\varphi)\right| < \frac{1}{n^{4c}}, \tag{3}$$

since for each $k \in \{0,1\}^{\kappa(n)}$ the probability of picking h_k from H is at least $1/2^{\kappa(n)} > 1/2^{n^\alpha}$.

From A we create the decision circuit Υ which accepts the function oracle w iff $A^w \geq \Pr_{f\in\mathcal{F}}(C^f) + 1/n^{3c}$. The following claim shows that Υ will accept an ε-fraction of the functions from H.

Claim. *For all sufficiently large n,*

$$\Pr_{h\in\mathsf{H}^n}\left[A_n^h \geq \Pr_{f\in\mathcal{F}^n}(C_n^f) + \frac{1}{n^{3c}}\right] \geq \varepsilon(n) + \frac{1}{n^{3c}}.$$

Proof. Again we fix n so that it is sufficiently large such that all of the inequalities below hold. Assume for contradiction that $\Pr_{h\in\mathsf{H}}[A^h \geq \Pr_{f\in\mathcal{F}}(C^f) + 1/n^{3c}] < \varepsilon(n) + 1/n^{3c}$. Let $\mathcal{T} \subseteq \mathsf{H}$ be the set of functions $h \in \mathsf{H}$, for which $A^h \geq \Pr_{f\in\mathcal{F}}(C^f) + 1/n^{3c}$, and let $\overline{\mathcal{T}}$ be its complement.

$$\begin{aligned}
\Pr_{\varphi\in\mathsf{G}\diamond\mathsf{H}}(C^\varphi) - \Pr_{f\in\mathcal{F}}(C^f) \;=\; & \sum_{h\in\mathcal{T}}\left(\left(\Pr_{\varphi\in\mathsf{G}\diamond h}(C^\varphi) - \Pr_{f\in\mathcal{F}}(C^f)\right)\Pr_{\psi\in\mathsf{H}}[\psi = h]\right) \\
& + \sum_{h\in\overline{\mathcal{T}}}\left(\left(\Pr_{\varphi\in\mathsf{G}\diamond h}(C^\varphi) - \Pr_{f\in\mathcal{F}}(C^f)\right)\Pr_{\psi\in\mathsf{H}}[\psi = h]\right) \\
\leq\; & \sum_{h\in\mathcal{T}}\left(\left(\Pr_{\varphi\in\mathsf{G}\diamond h}(C^\varphi) - \Pr_{f\in\mathcal{F}}(C^f)\right)\Pr_{\psi\in\mathsf{H}}[\psi = h]\right) \\
& + \sum_{h\in\overline{\mathcal{T}}}\left(\left(\left(A^h - \Pr_{f\in\mathcal{F}}(C^f)\right) + \frac{1}{n^{3c}}\right)\Pr_{\psi\in\mathsf{H}}[\psi = h]\right) \\
\leq\; & \sum_{h\in\mathcal{T}}\left(\left(\Pr_{\varphi\in\mathsf{G}\diamond h}(C^\varphi) - \Pr_{f\in\mathcal{F}}(C^f)\right)\Pr_{\psi\in\mathsf{H}}[\psi = h]\right) \\
& + \frac{1}{n^{3c}}
\end{aligned} \tag{4}$$

$$\leq \left(\varepsilon(n) + \frac{1}{n^{3c}}\right)\left(\delta(n) + \frac{1}{n^c}\right) + \frac{1}{n^{3c}} \tag{5}$$

$$\leq \varepsilon(n)\delta(n) + \frac{1 - 1/n^{c/2}}{n^c} + \frac{3}{n^{3c}} \tag{6}$$

$$< \varepsilon(n)\delta(n) + \frac{1}{n^c} \qquad \text{(contradiction)}. \tag{7}$$

Equation (4) follows from two facts. First, by assumption, the probability that a random $h \in \mathsf{H}$ is in $\overline{\mathcal{T}}$ is at most 1. Second, for each $h \in \overline{\mathcal{T}}$, $A^h - \Pr_{f\in\mathcal{F}}(C^f) < 1/n^{3c}$. Equation (5) also follows from two facts. First, by assumption, $\Pr_{h\in\mathsf{H}}[h \in \mathcal{T}] < \varepsilon(n) + 1/n^{3c}$. Second, by (2), for each $h \in \mathsf{H}$, $\Pr_{\varphi\in\mathsf{G}\diamond h}(C^\varphi) - \Pr_{f\in\mathcal{F}}(C^f) < \delta(n) + 1/n^c$. Equation (6) follows from the facts that $\varepsilon(n) \leq 1 - 1/n^{c/2}$ and $\delta \leq 1$. Equation (7) contradicts the fact that $\Pr_{\varphi\in\mathsf{G}\diamond\mathsf{H}}(C^\varphi) - \Pr_{f\in\mathcal{F}}(C^f) \geq \varepsilon(n)\delta(n) + 1/n^c$. □

Putting together the previous claim with Lemma 5, it is clear that Υ accepts an ε-fraction of the functions in H and a negligible fraction of random functions. The technical details are presented below:

$$\begin{aligned}
\Pr_{h\in\mathsf{H}}(\Upsilon^h) - \Pr_{f\in\mathcal{F}}(\Upsilon^f) &\geq \varepsilon(n) + \frac{1}{n^{3c}} - \Pr_{f\in\mathcal{F}}(\Upsilon^f) \\
&\geq \varepsilon(n) + \frac{1}{n^{3c}} - \frac{1}{2^{n^\alpha}} \\
&\quad - \Pr_{w\in\mathcal{F}}\left[\Pr_{g\in\mathsf{G}\diamond w}(C^g) \geq \Pr_{f\in\mathcal{F}}(C^f) + \frac{1}{n^{3c}} - \frac{1}{n^{4c}}\right] \qquad (8) \\
&\geq \varepsilon(n) + \frac{1}{n^{3c}} - \frac{1}{n^{4c}} - \frac{1}{2^{n^\alpha}} \qquad (9) \\
&\geq \varepsilon(n) + \frac{1}{n^{4c}} \qquad \text{(for sufficiently large } n\text{)}.
\end{aligned}$$

Equation (8) follows as A^ω approximates $\Pr_{g\in\mathsf{G}\diamond\omega}(C^g)$ to within a factor of $1/n^{3c}$ for all but $1/2^{n^\alpha}$ of the $\omega \in \mathcal{F}$. Equation (9) follows by a direct application of Lemma 5.

By constructing, for all applicable n, the circuit Υ_n in a straightforward manner and using standard circuit derandomization techniques, we can see that there exists a fixed polynomial p, for which the size of Υ_n is bound by $p(n^c \cdot c_{\mathsf{G}}(n))s_C(n)$.

3.5. *Proof of Lemma* 5

We begin with an outline of the proof of Lemma 5. We note that the essence of the proof lies in showing that for almost every function f, polynomial-sized circuits cannot distinguish between a function chosen randomly from $\mathcal{F}^n$ and a function chosen randomly from $\mathsf{I}(f) = \{f^r_\oplus \mid r \in \{0,1\}^n\}$. It follows that for a randomly chosen f from $\mathcal{F}^n$ the same circuits are not able to distinguish between random functions and those in $\mathsf{G}\diamond f$, proving the lemma.

The majority of work in this section goes into showing that for almost all functions f, polynomial-sized circuits cannot distinguish between functions chosen randomly from

$\mathcal{F}^n$ or $\mathrm{I}(f)$. A similar observation has previously been made by Even and Mansour [4] in the context of the construction of PRPGs from a random permutation oracle.

This result is proven by demonstrating that there is a random variable that, with very high probability, is a good approximation to both the probability that a circuit C accepts a random function and the probability that the circuit accepts a function from $\mathrm{I}(f)$. The random variable is an approximation of the value $\Pr_{g\in\mathrm{I}(f)}(C^g)$. This approximation is calculated by taking the average acceptance rate of C on a polynomial number of functions chosen randomly from $\mathrm{I}(f)$. It is the case that, *with high probability*, we can simulate many executions of C, given functions chosen uniformly at random from $\mathrm{I}(f)$ by simply executing C and responding to its oracle queries with random strings. Of course, one has to be careful that given two oracles $g_1, g_2 \in \mathrm{I}(f)$, the responses to the oracle queries of C^{g_1} and C^{g_2} are not inconsistent with an underlying random f. However, because of the random offsets r_1 and r_2, where $g_i(x) = f^{r_i}_{\oplus}$, it is unlikely that such responses will be inconsistent, and therefore with high probability we can approximate $\Pr_{g\in\mathrm{I}(f)}(C^g)$. It is clear that the same process can be used to approximate $\Pr_{f\in\mathcal{F}}(C^f)$, and therefore the same value is a good approximation to both values of interest and therefore the difference between the two values must be quite small.

Lemma 6. *Let $\{C_n\}$ be a family of polynomial in n sized decision-circuits. Then for any $d > 0$, there exists an $r > 0$ such that for all sufficiently large n and every $s \in \mathcal{F}^n$,*

$$\Pr_{(f,k_1,\ldots,k_{n^r})\in\mathcal{F}^n\times\{0,1\}^{(n^{2r})}}\left[\left|\frac{1}{n^r}\sum_{i=1}^{n^r} C_n^{f^{k_i}_{\oplus}} - \Pr_{f\in\mathcal{F}^n}(C_n^f)\right| > \frac{1}{n^d}\right] < \frac{1}{2^{n/3}}.$$

We remind the reader that we have modified all our circuits so that they will never repeat oracle queries, and each circuit C_n performs exactly $m(n)$ queries.

Proof. We assume the n is sufficiently large for all inequalities in this proof to hold and that n is fixed. Accordingly, we drop the index n from our notation as described at the end of Section 3.2. We assume that r is fixed, and later we show how to determine r's value.

First we define two experiments. In the first experiment pick random $(f_1, f_2, \ldots, f^r_n) \in (\mathcal{F})^{n^{2r}}$, and evaluate $C(f_i)$ for each $i \in \{1, \ldots, n^r\}$. Define the event E_1 to be

$$\left|\frac{1}{n^r}\sum_{i=1}^{n^r} C^{f_i} - \Pr_{f\in\mathcal{F}}(C^f)\right| > \frac{1}{n^d}.$$

In the second experiment pick random $(f, k_1, \ldots, k_{n^r}) \in \mathcal{F} \times \{0, 1\}(n^{2r})$, and evaluate $C^{f^{k_i}_{\oplus}}$ for each i, where $1 \le i \le n^r$. Define the event E_2 to be

$$\left|\frac{1}{n^r}\sum_{i=1}^{n^r} C^{f^{k_i}_{\oplus}} - \Pr_{f\in\mathcal{F}}(C^f)\right| > \frac{1}{n^d}.$$

Using the Chernoff Bound (Lemma 1) we choose an r where the probability of event E_1 occurring in the first experiment is less than $1/2^{n/2}$. We will show that the probability

of event E_2 in the second experiment, is negligibly close to the probability of E_1 in the first experiment.

We perform a third experiment in which we model both of the first two experiments. This is done by considering two different methods of evaluating the circuit C, n^r times.

We choose $\gamma_1^1, \ldots, \gamma_{m(n)}^1, \ldots, \gamma_1^{n^r}, \ldots, \gamma_{m(n)}^{n^r} \in \{0,1\}^{n \cdot m(n) \cdot n^r}$ and $k_1, \ldots, k_{n^r}, \in \{0,1\}^{n^r}$. Let g_j^i represent the jth oracle-gate of C when evaluating f_i in experiment one, or $f_\oplus^{k_i}$ in experiment two. Let I_j^i be the input to g_j^i in the experiment and let O_j^i be its output. We consider the gates in the following order: $g_1^1, g_2^1, \ldots, g_{m(n)}^1, \ldots, g_1^{n^r}, \ldots, g_{m(n)}^{n^r}$.

We model the first experiment of performing n^r evaluations of C with random functions by equivalently considering n^r evaluations of C, where in the evaluation of f_j we independently assign the element γ_i^j to O_j^i. Notice that by fixing, for each i and each j, the value of O_j^i we have completely determined the behavior of the circuit C in each of the evaluations. Further, notice that because of the way we have determined the outputs of the gates, we have modeled experiment one. Let $\mathcal{E}_1$ be the event in experiment three that corresponds to event E_1 in experiment one.

We introduce some notation to help us explain the second experiment. When we perform a query x on the oracle gate $f_\oplus^{k_i}$, then we say that query *evaluates* f on the input $x \oplus k_i$. Observe that we could model the second experiment in the same manner as the first, if we could guarantee that all of the queries to the oracle-gates result in f being *evaluated* only on distinct inputs. Unfortunately, in our experiment there is the possibility for $i \neq j$ that C will perform a query α on oracle $f_\oplus^{k_i}$ and a query β on oracle $f_\oplus^{k_j}$, where $\alpha \oplus k_i^2 = \beta \oplus k_j^2$, and the result will be that two different oracle gates *evaluate* f on the same input. In this case we cannot mimic the outputs of the two oracle gates with random strings of bits, as their responses will be inconsistent with respect to f, and therefore we will have failed to model experiment two. Fortunately, this is very unlikely to occur and therefore we can model experiment two in a method similar to which we modeled experiment one. First note that when given two pairs (a, b) and (c, d), where $a, b, c, d \in \mathbb{N}$, we say that $(a, b) < (c, d)$ iff $a < c$ or $a = c$ and $b < d$. We now describe the simulation of the second experiment in the third experiment.

In order to simulate, in experiment three, the evaluation of the circuits $C^{f_\oplus^{k_1}}, \ldots, C^{f_\oplus^{k_{n^r}}}$ in experiment two, we evaluate the circuit C, n^r times, as will be described. For the ith evaluation of C in the third experiment, which is simulating the evaluation of $C^{f_\oplus^{k_i}}$, we begin by setting the output of oracle gates O_ℓ^i to be γ_ℓ^i, where $1 \leq \ell \leq m(n)$. Observe that once the outputs of the oracle gates have been fixed then all of the inputs to the oracle gates are fixed as is the output of the circuit. We have now fixed the input/output pairs to each of the oracle gates. Unfortunately, we cannot be sure that this set of input/output pairs properly simulates input/output pairs that are consistent with the previous simulations of $C^{f_\oplus^{k_1}}, \ldots, C^{f_\oplus^{k_{i-1}}}$. The reason is that we can view $k_1, \ldots, k_i$ and the fixed input/output pairs that have been established in the experiment as partially defining the random function f, and so two oracle queries can be *evaluating* this partially defined f on the same input, but with inconsistent output. The only way that two oracle queries can cause such an f to be *evaluated* on the same input is if there are bad choices for our k_i's. For each i we determine if there exists a pair (a, b) and a j, where $(a, b) < (i, j)$, such that

$I^i_j \oplus k_i = I^a_b \oplus k_a$; if such an (a, b) and j exist, then we say a *collision* has occurred. A collision corresponds to a bad choice of k_i, and therefore f is inadvertently being *evaluated* twice on the same input, and therefore we have to make sure that the two query responses are consistent.

In order to ensure consistency, if a collision has occurred during the simulation of the evaluation of the $C^{f^{k_i}_\oplus}$, then we must re-evaluate it before continuing to simulate the evaluation of $C^{f^{k_{i+1}}_\oplus}$. We re-evaluate the gates $g^i_1, \ldots, g^i_{m(n)}$ in the specified order. For each gate g^i_j we consider its input I^i_j and the value that f is correspondingly evaluated on, $I^i_j \oplus k_i$. If there exists a pair $(a, b) < (i, j)$ such that $I^a_b \oplus k_a = I^i_j \oplus k_i$, then f is being evaluated on an input that it has previously been evaluated on. Therefore, we set the output O^i_j to be O^a_b and this forces the oracle gates to respond consistently with respect to f. If no such pair (a, b) existed, then the random response was a consistent response, and therefore we permit the original $O^i_j \leftarrow \gamma^i_j$. Observe that this method of evaluating C models experiment two. Let $\mathcal{E}_2$ be the event in experiment three that corresponds to E_2 in experiment two.

Note that the model of the second experiment is identical to the model of the first except in those cases in which a collision occurs. We define $\mathcal{E}_3$ to be the event that a collision occurred during the third experiment. Clearly $\mathcal{E}_2 \subseteq \mathcal{E}_1 \cup \mathcal{E}_3$, which implies that $\Pr(\mathcal{E}_2) \leq \Pr(\mathcal{E}_1) + \Pr(\mathcal{E}_3)$. Since the probability of E_1 is less than $1/2^{n/2}$ the probability of $\mathcal{E}_1$ is less than $1/2^{n/2}$. Therefore, it suffices to show that the probability of event $\mathcal{E}_3$ is less than $1/2^{n/2}$ in order to prove that the probability of $\mathcal{E}_2$ is less than $1/2^{n/3}$. This implies that the probability of E_2 is less than $1/2^{n/3}$, and this proves the lemma.

We bound from above the probability of event $\mathcal{E}_3$. We note that during the simulation of the evaluation of $C^{f^{k_i}_\oplus}$ there are at most $m(n)^2(i-1)$ possible values of k_i that would cause a collision during the evaluation of the circuit. This can be observed by noting that, for each query to a gate during the simulation of the evaluation of $C^{f^{k_i}_\oplus}$, there are $m(n)(i-1)$ choices of k_i that will cause a collision with the gate evaluations in the $(i-1)$ previously evaluated circuits, and there are $m(n)$ gates per circuit. Therefore, the probability of having a collision during the simulation of the evaluation of $C^{f^{k_i}_\oplus}$ is at most $((m(n))^2 \cdot (i-1))/2^n$. We note that the probability of a collision occurring in experiment three is less than the sum of the probabilities of a collision occurring during the simulation of the evaluations of $C^{f^{k_i}_\oplus}$ for each i. Therefore,

$$\begin{aligned}
\Pr[E_3] &\leq \frac{(m(n))^2}{2^n}(1 + 2 + \cdots + (n^r - 1)) \\
&= \left(\frac{(m(n))^2}{2^n}\right)\left(\frac{(n^r - 1)(n^r)}{2}\right) \\
&= \frac{(m(n))^2 n^{2r} - n^r (m(n))^2}{2^{n+1}} \\
&\leq \frac{1}{2^{n/3}} \qquad \text{(for sufficiently large } n\text{).}
\end{aligned}$$

□

Lemma 7. *Let $\{C_n\}$ be a polynomial-sized family of decision-circuits. Then for every $s \in \mathcal{F}^n$, for every constant c, for sufficiently large n and for all but $1/2^{n/4}$ of the $w \in \mathcal{F}^n$,*

$$\left| \Pr_{f \in s \diamond w}(C_n^f) - \Pr_{g \in \mathcal{F}^n}(C_n^g) \right| < \frac{1}{n^c}.$$

Proof. The intuition behind this proof is as follows. We use Lemma 6 to establish with high probability for random w and k_i that $(1/n^u)\sum_{i=1}^{n^u} C_n^{w^{k_i}_{\oplus}}$ is a good approximation of $\Pr_{g \in \mathcal{F}}(C^g)$. Then, by the Chernoff Bound, the same value is a good approximation to $\Pr_{r \in \{0,1\}^n}(C^{w^r_{\oplus}})$. This implies that for almost all appropriate functions w the values $\Pr_{g \in \mathcal{F}}(C^g)$ and $\Pr_{r \in \{0,1\}^n}(C^{w^r_{\oplus}})$ are close. Next, we derive that for any fixed s, for almost all w, the values $\Pr_{g \in \mathcal{F}}(C^g)$ and $\Pr_{g \in s \diamond w}(C^g)$ are close, proving the theorem.

Let $\varphi(u)$ be shorthand for $(1/n^u)\sum_{i=1}^{n^u} C_n^{w^{k_i}_{\oplus}}$. By Lemma 6 we know that there exists an r_0 such that $\forall r > r_0$ and for all sufficiently large n,

$$\Pr_{(w,k_1,\ldots,k_{n^r}) \in \mathcal{F}^n \times \{0,1\}^{n^{2r}}} \left[\left| \varphi(r) - \Pr_{g \in \mathcal{F}^n}(C_n^g) \right| > \frac{1}{n^{2c}} \right] < \frac{1}{2^{n/3}}. \tag{10}$$

By the Chernoff Bound we know that there exists a t_0 such that $\forall t > t_0$ and for all sufficiently large n,

$$\Pr_{(w,k_1,\ldots,k_{n^t}) \in \mathcal{F}^n \times \{0,1\}^{n^{2t}}} \left[\left| \varphi(t) - \Pr_{r \in \{0,1\}^n}(C_n^{w^r_{\oplus}}) \right| > \frac{1}{n^{2c}} \right] < \frac{1}{2^n}.$$

Let $v = \max\{t_0, r_0\}$, and consider the probability that either of the events occurs:

$$\begin{aligned} \Pr_{(w,k_1,\ldots,k_{n^v}) \in \mathcal{F}^n \times \{0,1\}^{n^{2v}}} \left[\left| \varphi(v) - \Pr_{r \in \{0,1\}^n}(C_n^{w^r_{\oplus}}) \right| \right. &> \frac{1}{n^{2c}} \bigvee \left. \left| \varphi(v) - \Pr_{g \in \mathcal{F}^n}(C_n^g) \right| > \frac{1}{n^{2c}} \right] \\ &< \frac{1}{2^{n/3}} + \frac{1}{2^n}. \end{aligned}$$

It follows for all sufficiently large n that

$$\Pr_{w \in \mathcal{F}^n} \left[\left| \Pr_{r \in \{0,1\}^n}(C_n^{w^r_{\oplus}}) - \Pr_{g \in \mathcal{F}^n}(C_n^g) \right| > \frac{1}{n^c} \right] < \frac{1}{2^{n/4}}.$$

Fix $s \in \mathcal{F}^n$. It follow that

$$\Pr_{w \in \mathcal{F}^n} \left[\left| \Pr_{r \in \{0,1\}^n}(C_n^{s \oplus w^r_{\oplus}}) - \Pr_{g \in \mathcal{F}^n}(C_n^g) \right| > \frac{1}{n^c} \right] < \frac{1}{2^{n/4}}.$$

This can be rewritten as

$$\Pr_{w \in \mathcal{F}^n} \left[\left| \Pr_{r_1, r_2 \in \{0,1\}^n}(C_n^{s^{r_1}_{\oplus} \oplus w^{r_2}_{\oplus}}) - \Pr_{g \in \mathcal{F}^n}(C_n^g) \right| > \frac{1}{n^c} \right] < \frac{1}{2^{n/4}}.$$

Which can be rewritten as

$$\Pr_{w\in\mathcal{F}^n}\left[\left|\Pr_{g\in s\diamond w}(C_n^g)-\Pr_{g\in\mathcal{F}^n}(C_n^g)\right|>\frac{1}{n^c}\right]<\frac{1}{2^{n/4}},$$

proving the claim. □

The final lemma now follows from Lemma 6 and a simple averaging argument.

Lemma 8 (Restatement of Lemma 5). *For $i>0$ and for each n let*

$$K_n(i)=\Pr_{f\in\mathcal{F}^n}(C_n^f)+\frac{1}{n^i}\quad \textit{and let}\quad S^n(i)=\left\{w\in\mathcal{F}^n\,\middle|\,\Pr_{g\in\mathsf{G}^n\diamond w}(C_n^g)\geq K_n(i)\right\}.$$

Then for all i,j, $\Pr_{w\in\mathcal{F}^n}(w\in S^n(i))\leq 1/n^j$, for sufficiently large n.

Proof. Suppose for contradiction that there exists an i and j such that for infinitely many n, $\Pr_{w\in\mathcal{F}^n}(w\in S^n(i))\geq 1/n^j$. We show this contradicts Lemma 7. We fix n and drop the index from the notation. Note that since $\Pr_{\varphi\in\mathsf{G}\diamond S(i)}(C^\varphi)\geq\Pr_{f\in\mathcal{F}}(C^f)+1/n^i$, then by an averaging argument we can fix a $g\in\mathsf{G}$ such that $\Pr_{h\in g\diamond S(i)}(C^h)\geq\Pr_{f\in\mathcal{F}}(C^f)+1/n^i$. Using the first moment method we note that given g there must be a $1/n^{2i}$ fraction of $w\in S(i)$ that have the "good" property that $\Pr_{\psi\in g\diamond w}(C^\psi)\geq\Pr_{f\in\mathcal{F}}(C^f)+1/n^{2i}$. Since $\Pr_{w\in\mathcal{F}}(w\in S(i))\geq 1/n^j$, the probability that a random w has the "good" property is $1/n^{2i+j}$, and this contradicts Lemma 7. □

4. Discussion and Further Research

We have presented a relatively simple and efficient construction for transforming a partially secure PRFG into a strongly secure PRFG. We believe this construction could possibly be used to guide the development of block-ciphers in the future. However, the construction may be useful only in outer layers of the cipher, after a certain minimal amount of security has been achieved by other means.

4.1. *Some Observations on the Uniform Adversarial Model*

The proofs in this paper have been presented with respect to a non-uniform adversary. As referred to earlier, the same proof can be re-worked to hold with respect to a uniform adversary. We present some of the more important points. First, we re-state the Isolation Lemma with respect to polynomial-time constructible probabilistic circuits, as opposed to non-uniform circuit families. We present the uniform version below:

Lemma 9 (Uniform Version of Diamond Isolation Lemma). *Let p,ε,δ and d be as they were in Lemma* 2. *Let* H *and* G *be function generators* whose circuits can be constructed in polynomial-time, *where $c_{\mathsf{G}}(n)$ and $c_{\mathsf{H}}(n)$ are polynomials that bound from above the size of the circuits that compute the function generators respectively.*

Hypothesis: *Let s_C be a polynomial. There exists a family of polynomial-time constructible, decision-circuits $\{C_n\}$, where for each n the circuit C_n is of size bounded above by $s_C(n)$, and there exists $c > 2d$ such that for infinitely many n,*

$$\left| \Pr_{g \in \mathsf{G}^n \diamond \mathsf{H}^n}(C_n^g) - \Pr_{f \in \mathcal{F}^n}(C_n^f) \right| \geq \varepsilon(n)\delta(n) + \frac{1}{n^c}.$$

Conclusion: *There exists a family of polynomial-time computable, probabilistic, decision-circuits such that for infinitely many n either the circuit family is composed of circuits Υ_n of size $p(n^c \cdot c_{\mathsf{G}}(n))s_C(n)$ for which*

$$\left| \Pr_{h \in \mathsf{H}^n}(\Upsilon_n^h) - \Pr_{f \in \mathcal{F}^n}(\Upsilon_n^f) \right| \geq \varepsilon(n) + \frac{1}{n^{3c}},$$

or circuits Ξ_n of size $\leq (2Q_{C_n}c_{\mathsf{H}}(n) + s_C(n))$, where $Q_{\Xi_n} = Q_{C_n}$, and for which

$$\left| \Pr_{g \in \mathsf{G}^n}(\Xi_n^g) - \Pr_{f \in \mathcal{F}^n}(\Xi_n^f) \right| \geq \delta(n) + \frac{1}{n^{c+1}}.$$

The important differences between the uniform and non-uniform versions of the Isolation Lemma are: firstly, in the uniform version G and H must be uniformly constructible; and, secondly, the distinguishing probability in the latter half of the conclusion is weaker ($\delta(n) + 1/n^c$ versus $\delta(n) + 1/n^{c+1}$). The need for the first difference is clear: the proof of the lemma requires that we be able to compute G and H, and this requires that they be uniformly constructible. The only other part of the proof of the Isolation Lemma that blatantly uses non-uniformity is Lemma 4. It is the uniform version of this lemma which leads to the weaker distinguishing probability in the uniform version of the Isolation Lemma. We state and sketch the proof for the uniform version of Lemma 4.

Lemma 10. *Either there exists a family of decision-circuits $\{\Xi_n\}$* whose circuits can be constructed with high probability in probabilistic polynomial-time, *where for each n the circuit Ξ_n is of size no greater than $Q_{C_n}2c_{\mathsf{H}}(n) + s_C(n)$; $Q_{\Xi_n} = Q_{C_n}$; and for infinitely many n,*

$$\left| \Pr_{g \in \mathsf{G}^n}(\Xi_n^g) - \Pr_{f \in \mathcal{F}^n}(\Xi_n^f) \right| \geq \delta(n) + \frac{1}{n^{c+1}};$$

or for all sufficiently large n and a $1 - 1/n^{3c}$ fraction of $h \in \mathsf{H}^n$,

$$\left| \Pr_{g \in \mathsf{G}^n \diamond h}(C_n^g) - \Pr_{f \in \mathcal{F}^n}(C_n^f) \right| < \delta(n) + \frac{1}{n^c}.$$

(*We remind the reader that Q_C represents the number of oracle queries performed by circuit C.*)

Proof. (Sketch) Suppose it is the case that for infinitely many n there exists at least a $1/n^{3c}$ fraction of $h \in \mathsf{H}^n$ for which $|\Pr_{g \in \mathsf{G} \diamond h}(C_n^g) - \Pr_{f \in \mathcal{F}^n}(C_n^f)| \geq \delta(n) + 1/n^c$. For

such n call the $h \in \mathsf{H}$ for which this holds *useful*. If for each such n we could efficiently find a *useful* h, then we could construct a decision circuit Ξ_n, defined as $\Xi_n^w = C_n^{(w \diamond h)}$, and (as in the non-uniform version)

$$\begin{aligned}\left| \Pr_{\psi \in \mathsf{G}^n}(\Xi_n^\psi) - \Pr_{f \in \mathcal{F}^n}(\Xi_n^f) \right| &= \left| \Pr_{\psi \in \mathsf{G}^n \diamond h}(C_n^\psi) - \Pr_{f \in \mathcal{F}^n \diamond h}(C_n^f) \right| \\ &= \left| \Pr_{\psi \in \mathsf{G}^n \diamond h}(C_n^\psi) - \Pr_{f \in \mathcal{F}^n}(C_n^f) \right| \qquad \text{(Lemma 3)} \\ &\geq \delta(n) + \frac{1}{n^c},\end{aligned}$$

proving the lemma. Unfortunately, in the uniform model we cannot just choose such an h: it must be found efficiently. In actuality one is found with all but negligible probability, but this will suffice.

Such an $h \in \mathsf{H}^n$ is found by sampling. More specifically: for a random $h \in \mathsf{H}^n$ we will approximate the value of $|\Pr_{g \in \mathsf{G}^n \diamond h}(C_n^g) - \Pr_{f \in \mathcal{F}^n}(C_n^f)|$ to within an accuracy of $1/n^{2c}$, with all but negligible probability. If the value is greater than $\delta(n) + 2/n^{c+1}$, we say that h is *good* and use it in the construction, otherwise we reject it. Observe that if a random h is *useful*, then with all but negligible probability it will be found to be *good*. Therefore, if, for the given n, a $1/n^{3c}$ fraction of the $h \in \mathsf{H}$ are *useful*, then, by the Chernoff Bound, with all but negligible probability after $\mathcal{O}(n^{3c})$ selections of random h's we will find a *good* h. If after the $\mathcal{O}(n^{3c})$ selections no *good* h has been found, then a random h is used in the construction. For all n for which a $1/n^{3c}$ fraction of the $h \in \mathsf{H}^n$ are useful, it follows that

$$\begin{aligned}\left| \Pr_{\psi \in \mathsf{G}^n}(\Xi_n^\psi) - \Pr_{f \in \mathcal{F}^n}(\Xi_n^f) \right| &= \left| \Pr_{\psi \in \mathsf{G}^n \diamond h}(C_n^\psi) - \Pr_{f \in \mathcal{F}^n \diamond h}(C_n^f) \right| \\ &= \left| \Pr_{\psi \in \mathsf{G}^n \diamond h}(C_n^\psi) - \Pr_{f \in \mathcal{F}^n}(C_n^f) \right| \qquad \text{(Lemma 3)} \\ &\geq \delta(n) + \frac{1}{n^{c+1}} \qquad \text{(with probability} \approx 1 \text{ and large } n).\end{aligned}$$

It remains to explain how to approximate $|\Pr_{g \in \mathsf{G} \diamond h}(C_n^g) - \Pr_{f \in \mathcal{F}^n}(C_n^f)|$ to within an accuracy of $1/n^{2c}$ with all but negligible probability. We observe that this value can be approximated, as is required in the proof above, by appropriate sampling, as is done in the proof of Lemma 6.

Finally we stress that, while it takes a great deal (but still polynomial time) of simulation to determine the appropriate h to use in the construction of Ξ_n, it is *still* the case that the size of Ξ_n is no greater than $s_C(n) + Q_{C_n}(2C_{\mathsf{H}}(n))$ and the number of oracle gates remains $Q_{\Xi_n} = Q_{C_n}$. □

The remainder of the proof of the Isolation Lemma goes through almost unchanged. The only minor change being that now in the proof of the claim in Section 3.4, rather

than $\Pr_{\varphi \in \mathsf{G} \diamond h}(C^{\varphi}) - \Pr_{f \in \mathcal{F}}(C^{f}) < \delta(n) + 1/n^{c}$ holding for each $h \in \mathsf{H}$, it only holds for a $1 - 1/n^{3c}$ fraction of the $h \in \mathsf{H}$. This modifies the probability calculations by an additive term of $1/n^{3c}$, but there is enough slack in the inequalities involved in the claim that it does not affect the result of the claim.

Finally, a new version of Theorem 1 can be proven by taking into account the slightly weaker distinguishing probabilities achieved by the uniform version of the Isolation Lemma.

4.2. *Future Research and Open Problems*

As has been previously mentioned, there are significant similarities between this proof and Levin's proof of the XOR Lemma. Further, as was previously mentioned, there are several different proofs of Yao's XOR Lemma. Given the parallels between Levin's result and our own, it is interesting to question if any of the other proof techniques used to prove the XOR Lemma can be used to prove security amplification in the context of a weak PRFG.

For example, it is interesting to ask if there is some natural notion for a weak PRFG that corresponds to hard-core sets of weakly unpredictable predicates, as proposed by Impagliazzo [9]. One natural notion for a hard-core set of weak PRFGs is to consider subsets of keys for the generator that correspond to "random" looking functions. However, this notion does not seem to work well as there are examples of weak PRFGs that have no hard-core sets using this notion. Consider the sketched construction used in Section 3.1. Every subset of keys of this generator will correspond to a set of functions that can be distinguished with probability at least $\frac{1}{2}$, and thus are not very "random". Of course, there might be another notion of a hard-core set that is more applicable to weak PRFGs.

We believe it is an interesting open question to determine if any of the other proofs of the XOR Lemma have natural analogies for security amplification of weak PRFGs.

Acknowledgments

The author thanks Charles Rackoff for suggesting the problem and for many valuable discussions and suggestions. The author also thanks the referees. Their suggestions have greatly improved the presentation of this paper.

References

[1] W. Aiello, M. Bellare, G. Di Crescenzo, and R. Vekatesan. Security amplification by composition: the case of doubly-iterated, ideal ciphers. In H. Krawczyk, editor, *Advances in Cryptology - Crypto* 98, volume 1462 of LNCS, pages 390–407. Springer-Verlag, Berlin, 1998.

[2] K. Akcoglu and C. Rackoff. Pseudo-random permutation generators and cryptographic composition. Draft manuscript.

[3] J. Daemen, R. Govaerts, and J. Vandewalle. Weak keys for IDEA. In *Advances in Cryptology — CRYPTO* 93 *Proceedings*, volume 773 of LNCS, pages 224–232. Springer-Verlag, Berlin, 1993.

[4] S. Even and Y. Mansour. A construction of a cipher from a single pseudorandom permuation. In H. Imai, R. L. Rivest, and T. Matsumoto, editors, *Advances in Cryptology*, *ASIACRYPT* '91: *International Conference on the Theory and Application of Cryptology*, *Fujiyoshida*, *Japan*, *November* 11–14, 1991, *Proceedings*, volume 739 of LNCS, pages 210–224. Springer-Verlag, Berlin, 1993.

[5] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986.
[6] O. Goldreich, N. Nisan, and A. Wigderson. On Yao's xor-lemma. http://theory.lcs.mit.edu/~oded/, 1995.
[7] J. Hastad, R. Impagliazzo, L. A. Levin, and M. Luby. Construction of pseudorandom generator from any one-way function. *SIAM Journal of Computing*, 28(4):1364–1396, 1998.
[8] R. Impagliazzo. Hard-core distributions for somewhat hard problems. http://www-cse.ucsd.edu/~russell/, 1994.
[9] R. Impagliazzo. Hard-core distributions for somewhat hard problems. In *Proceedings of the* 36*th Annual Symposium on Foundations of Computer Science*, *October* 23–25, 1995, *Milwaukee*, *Wisconsin*, pages 538–545. IEEE Computer Society Press, Los Alamitos, CA, 1995.
[10] J. Kilian and P. Rogaway. How to protect DES against exhaustive key search. In N. Koblitz, editor, *Advances in Cryptology — Crypto* 96, volume 1109 of LNCS, pages 252–267. Springer-Verlag, Berlin, 1996.
[11] L. A. Levin. One-way functions and pseudorandom generators. *Combinatorica*, 7(4):357–363, 1987.
[12] M. Luby and C. Rackoff. Pseudo-random permutation generators and cryptographic composition. In *Proceedings of the* 18*th Annual Symposium on Theory of Computing*, pages 353–363. ACM, New York, 1986.
[13] M. Luby and C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal on Computing*, 17:373–386, 1988.
[14] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, Cambridge, 1995.
[15] S. Myers. On the Development of Pseudo-Random Function Generators and Block-Ciphers Using the XOR and Composition Operators. M.Sc. Thesis, University of Toronto, 1999.
[16] M. Naor and O. Reingold. On the construction of pseudo-random permutations: Luby–Rackoff revisited. *Journal of Cryptology*, 12(1):29–66, 1999.
[17] A. Yao. Theory and applications of trapdoor functions (extended abstract). In *Proceedings of the* 23*rd Symposium on Foundations of Computer Science*, pages 80–91. IEEE, New York, 1982.