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

Steven Myers

myers@cs.toronto.edu
Department of Computer Science
University of Toronto
Toronto, Ontario, Canada

Abstract. We show that given a PRFG (pseudo-random function generator) G which is $\frac{1}{c}$-partially secure, the construction $g_1(x \oplus r_1) \oplus \cdots \oplus g_{\log^2 n}(x \oplus r_{\log^2 n})$ produces a strongly secure PRFG, where $g_i \in G$ and r_i are strings of random bits. Thus we present 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 "naturally" construct a PRPG from partially secure PRPG.

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 both considered to be secure (given its small key size) and conjectured to be a Pseudo-Random Permutation Generator(PRPG). This construction has 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 [8] defined the notion of a partially secure PRPG to be a permutation generator which produces permutations that cannot be efficiently distinguished from random permutations by small circuits with a probability better than $\frac{1}{c}$, 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 then 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 a partially secure PRPG implied a totally secure PRPG. The construction used the following chain of results. It is possible to construct a weak one-way function from a partially secure PRPG; then, using [13, 7], construct a one-way function; then, using Yao's XOR Lemma [4], construct a Pseudo-random number generator (PRNG); then, using [2], construct a PRFG; and then finally, using [9, 12], construct a PRPG. However, this construction is obviously neither "natural" nor efficient.

In this paper we give a natural, efficient and parallelizable construction for generating a Pseudo-Random Function Generator(PRFG) from a partially secure PRFG. Our proof follows from the ideas of Luby and Rackoff [8]. Further, since partially secure PRPG are a special case of partially secure PRFG, we can use a partially secure PRPG to construct a PRFG. Then, using a previous result by Luby and Rackoff [9], or more recent work by Naor and Reingold [12], we can "naturally" and efficiently construct a PRPG from the PRFG. If $\mathsf{F} = \{\mathsf{F}^n\}$ is a "partially secure" pseudo-random function generator, 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 PRFG. Further support for this this view is found in the fact that our proof closely follows that of Levin's in [7].

Given the relatively few number of proofs showing 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. Thus, while there are no natural examples of partially secure PRFG that the author is aware of, should cipher-designers develop efficient function generators which they have reason to believe are partially secure, then they can use the construction suggested in this paper, and have good reason to believe that the resulting cipher has stronger security properties than its constituents.

For the purposes of example only, suppose block-cipher designers had reason to believe that an 8-round version of DES was a "partially" secure PRFG[1]. Then designers could have some faith that the suggested construction could be used amplify the security of this "partially" secure generator. Further, the parallelizability of the construction might allow designers to make certain time/space trade-offs. For example, the designers might trade-off the time required for more rounds of DES, with the circuit size required to implement the above construction with a version of DES with fewer rounds.

1.1 Related Work

There are very few results in cryptography which demonstrate the amplification of security in a general, non-restrictive adversarial model. The first such result was Yao's XOR Lemma [13], which now has several proofs ([7, 5, 3]). All of these results apply to the security amplification of weak one-way functions and predicates. In a domain closer to that of PRFG, Luby and Rackoff [8] give a direct

[1] We use the quotes around "partially" as DES in not an asymptotic notion

product lemma for PRPG where the direct product is taken via the composition of weak PRPG. Unfortunately, their proof falls short of demonstrating that the direct product of a sufficient number of weak PRPG yields a a strongly secure PRPG. The reason for this is explained in further detail in the sequel. A direct product theorem for PRFG is given by Myers [11], where the direct product is based on the composition and exclusive-or of PRFG. Unfortunately, this result also fails to achieve a strongly secure PRFG for reasons similar to those of [8]. Further complicating the matters with the result in [11] is the fact that the size of the constructed generator is super-polynomial after $\omega(\log n)$ applications of the direct product.

Therefore, our result presents the first efficient and natural direct product theorem achieving strongly secure PRFG from weakly secure PRFG in a general adversarial model.

Since Luby and Rackoff proposed their partial security model in [8], cryptographers have developed other models where it is possible to demonstrate some manner of security amplification. Kilian and Rogaway [6] 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 to 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 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.

As previously alluded to, under this model Kilian and Rogaway [6] have shown that the DESX construction increases the effective key length of DES. Also under the same model, Aiello et al. [1] have shown that the composition of multiple random permutation generators results in a permutation generator which is more secure than a random generator.

2 Notation, Definitions & the Model

Below we introduce some notation and terminology which will be 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 $f : \{0,1\}^l \to \{0,1\}^p$, 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 *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 $\mathcal{D}_1, \mathcal{D}_2, ...$ be a sequence of distributions, and let e represent a series of events $e_1, e_2, ...$ 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 $\Pr_{\mathcal{D}_n}(e_n) \geq \frac{1}{n^c}$. We say that an event e occurs with* negligible probability *if, for all constants $c > 0$ and for all sufficiently large n, $\Pr_{\mathcal{D}_n}(e_n) < \frac{1}{n^c}$.*

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 which are used to define resource bounded adversaries: uniform and non-uniform. In this paper we will consider only non-uniform adversaries.

A non-uniform adversary is a sequence of circuits $(C_1, C_2, ...)$, 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 which 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.*

Notation 5 *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 6 *We write C^f to represent a circuit C that has oracle gates which compute the function f in unit time. We wish to consider these gates as "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 7 *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 shall assume that all circuits are standardized in the following manner: no circuit will ever repeat oracle queries, and all circuits C_n in a circuit family $\{C_n\}$ will perform exactly $m(n)$ queries, for some polynomial m (ie. $Q_{C_n} = m(n)$). Any polynomial sized family of circuits can easily be modified to satisfy the above two requirements.

2.2 Function Generators

Definition 5. *We call $G : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^m$ a function generator. We say that $k \in \{0,1\}^\kappa$ is a* key *of G, and we 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 $G^n : \{0,1\}^{\ell(n)} \times \{0,1\}^n \to \{0,1\}^{m(n)}$ be a function generator. We call $G = \{G^n | n \in \mathcal{N}\}$ a function generator ensemble.

In an abuse of notation, we will often refer to both specific function generators and function generator ensembles as function generators. We hope it will be clear from the context which term is actually being referred to.

Definition 6 (ϵ-Distinguishing Adversary). *Let $\epsilon : \mathbb{N} \to [0,1]$, and let $\mathcal{D}^1 = \{\mathcal{D}_i^1 | i \in \mathbb{Z}^+\}$ and $\mathcal{D}^2 = \{\mathcal{D}_i^2 | i \in \mathbb{Z}^+\}$ be two sequence of distributions over oracle gates, where $\mathcal{D}_i^j$ 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:*

$$\left| \Pr_{d_1 \in \mathcal{D}^1} \left[C_n^{d_1} = 1 \right] - \Pr_{d_2 \in \mathcal{D}^2} \left[C_n^{d_2} = 1 \right] \right| \geq \epsilon(n) + \frac{1}{p(n)}.$$

Definition 7 (Pseudo-Random Function Generator Ensembles). *Let m and ℓ be polynomials. For each n let $G^n : \{0,1\}^{\ell(n)} \times \{0,1\}^n \to \{0,1\}^{m(n)}$ be a* function generator*, computable in time bounded by a polynomial in n. Define $G = \{G^n | n \in \mathbb{N}\}$ to be the function generator ensemble. Define $\mathcal{F} = \{\mathcal{F}^{n \to m(n)} | n \in \mathbb{N}\}$.*

We say that G is $(1 - \epsilon(n))$ 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 $\frac{1}{p(n)}$ 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 Previously Known Lemmas

Below is a well known form of the Chernoff bound. For a proof of this result refer to [10] or any standard book on probabilistic computation.

Lemma 1 (Chernoff Bound). *Let $x_1, .., x_{n^t}$ be i.i.d.r.v. which take the values 0 or 1 with probabilities q or $p = 1 - q$ respectively. Let $X_{n^t} = \frac{1}{n^t} \sum_{i=1}^{n^t} x_i$. Then for any k and l, there exists a t such that:*

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

The following lemma is a generalization of standard derandomization proofs in the non-uniform computation model. Before stating the lemma, we give the following intuition of its statement. Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be two distribution over oracle functions, and P be a predicate with a domain over functions. Then if C is a probabilistic circuit such that $C^{\mathcal{D}_1}$ approximates $P(\mathcal{D}_1)$ and $C^{\mathcal{D}_2}$ approximates $P(\mathcal{D}_2)$, then there exists a derandomized version of C which approximates both $P(\mathcal{D}_1)$ and $P(\mathcal{D}_2)$.

Lemma 2 (Derandomization Lemma). *Let $C^w(r)$ be a probabilistic oracle-circuit, where w is an oracle function, and r is a string of random input bits. Let D^1 and D^2 be two distributions over $\mathcal{F}^n$, and let R be the distribution over C's random bits. Let $P : \mathcal{F}^n \times \mathbb{R} \to \{0,1\}$ be a predicate. Then, If*

$$\Pr_{w \in D^1,\, r \in R}[P(w, C^w(r)) = 1] \geq 1 - p \quad \textit{and} \quad \Pr_{w \in D^2,\, r \in R}[P(w, C^w(r)) = 1] \geq 1 - p,$$

then there exists an $\tilde{r} \in R$ such that $\Pr_{w \in D^i}[P(w, C^w(\tilde{r})) = 1] \geq 1 - 2p$, for $i \in \{1, 2\}$.

Proof. This result is a generalization of standard derandomization techniques for non-uniform circuits. The details are left to the full version of the paper. □

3 Result

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

Let f_1 and f_2 be two functions such that $f_i : \{0,1\}^n \to \{0,1\}^n$, for $i \in \{1,2\}$. For each $r_1, r_2 \in \{0,1\}^n$ we 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 type $\{0,1\}^n \to \{0,1\}^n$ 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 $\diamond = \{\diamond^n_{r_1,r_2} | n \in \mathbb{N} \wedge r_1, r_2 \in \{0,1\}^n\}$.

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

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

Similarly, if $g : \{0,1\}^n \to \{0,1\}^n$*, then we write* $g \diamond \mathsf{H}$ *as short-hand for the function generator defined by* $\mathsf{F}^n(k_2 \bullet k_3 \bullet k_4, x) = \left(g \diamond^n_{k_3 \bullet k_4} h^n_{k_2}\right)(x)$*, 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 $f^n \in \mathsf{F}$ it is sufficient to compute

$$\left(g_1(x \oplus r_1) \oplus \cdots g_{p(n)}(x \oplus r_n)\right),$$

where $g_i \in \mathsf{G}$ 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 are necessary for the security amplification, and a counter example to our security amplification claims can easily be constructed if they are omitted. For further discussion on this construction and several other plausible candidates see [11].

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 [7] in proving Yao's XOR Lemma [13]) stated below. Intuitively, the lemma shows that the function generator which 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. The majority of the work in this paper goes towards proving this lemma correct.

Lemma 3 (Diamond Isolation Lemma). *There exists a fixed polynomial* p *(which is retrievable form the proof of this lemma) such that the following hold. Let* $\epsilon, \delta : \mathbb{Z} \to [0,1]$ *be functions. 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: *There exists a family of decision-circuits* $\{C_n\}$*, where for each* n *the circuit* C_n *is of size bounded above by the polynomial* $s_C(n)$*, and for some*

$c > 0$ and infinitely many n:

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

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:*

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

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:

$$\left| \Pr_{g \in \mathsf{G}^n}(\Xi_n^g) - \Pr_{f \in \mathcal{F}^n}(\Xi_n^f) \right| \geq \delta(n) + \frac{1}{n^{2c}}.$$

Luby and Rackoff prove a similar lemma in [8]. It shows that the composition of two partially secure PRPGs results in a generator which is more secure than either of the constituents. Excluding the fact that their lemma is restricted to permutation generators instead of function generators, our lemma is stronger in two senses. First, the security requirement in the hypothesis is strictly weaker (ie. 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 which distinguishes $\mathsf{G} \circ \mathsf{H}$. It is this second fact that permits us to achieve PRFGs in our construction. Furthermore, this proof is simpler than that of Luby and Rackoff. Their proof contains a corollary which corresponds to Corollary 5 in our proof. However, unlike Corollary 5, 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 $1 - \epsilon$ secure PRFG.

Theorem 1 (Diamond Composition Theorem). *Let $0 < \epsilon < 1$ be a constant. Let G be a $1-\epsilon$ secure PRFG. Then for each $p \in \Omega(\log^2 n) \cap \left(\cup_{i=1}^{\infty} \mathcal{O}(n^i)\right)$ the generator $\mathsf{F} = \underbrace{\mathsf{G}\diamond \cdots \diamond \mathsf{G}}_{p(n)}$ is a secure PRFG.*

Proof. (sketch) The intuition for this argument is as follows. We assume that F is not secure, and thus there is a family of distinguishing circuits for F. We apply the Isolation Lemma to the generator F. The result is either that the generator G is not $1 - \epsilon$ secure as claimed, or we have a family of distinguishing circuits (slightly larger than the original circuit family) for a generator smaller than F. We apply the Isolation Lemma inductively to this smaller generator until we are only left with an $\epsilon + \frac{1}{n^c}$ family of distinguishing circuits for the generator G, which contradicts its assumed $1 - \epsilon$ security. The theorem follows. The full details are left to the full version of the paper. □

Before presenting a proof of the Diamond Isolation Lemma (Lemma 3), we first present two important technical lemmas. A complete proof of the Diamond Isolation Lemma follows.

3.2 Two Technical Lemmas

The first lemma and corollary demonstrate that the acceptance probability of an oracle-decision-circuit is the same whether the circuit is given an oracle chosen uniformly at random from the set of all functions; or given an oracle chosen uniformly at random from the set of all functions combined with any specific function using the $\diamond$ operator generator.

Lemma 4. *Given any decision-circuit C, for each $h \in \mathcal{F}^n$ and for each $r_1, r_2 \in \{0,1\}^n$:*

$$\Pr_{\phi \in h \diamond^n_{r_1 \bullet r_2} \mathcal{F}^n}(C^\phi) = \Pr_{f \in \mathcal{F}^n}(C^f).$$

Proof. First observe that for each $r_2 \in \{0,1\}^n$ the distribution $\{h'(x \oplus r_2) | h' \in \mathcal{F}^n\} = \mathcal{F}^n$. Then let $g(x) = h(x \oplus r_1)$, and observe that the distribution $g \oplus \mathcal{F}^n = \mathcal{F}^n$, proving the result.

Corollary 1. *Given any decision-circuit C, for each $h \in \mathcal{F}^n$:* $\Pr_{\phi \in h \diamond \mathcal{F}^n}(C^\phi) = \Pr_{f \in \mathcal{F}^n}(C^f)$.

The next lemma demonstrates that the probability of acceptance by a *polynomial sized* oracle-decision-circuit is "almost" the same whether given access to an oracle chosen uniformly at random from the set of all functions; or given an oracle chosen randomly from the set of functions specified by combining, via the $\diamond$ operator generator, any distribution of functions with "almost" any specific function.

Lemma 5. *Let $\{C_n\}$ be a polynomial sized family of decision-circuits. Then for every constant c, for sufficiently large n, for each $s \in \mathcal{F}^n$, for all but $\frac{1}{2^{n/4}}$ of the $w \in \mathcal{F}^n$:*

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

Proof. (sketch) Below we outline the high-level ideas behind the proof of the lemma. We leave the detailed proof for the full version of the paper.

In the remainder of this proof sketch, when we say a value has a good approximation, we imply it approximates the value to within a $\frac{1}{\text{poly}(n)}$-additive factor, where poly(n) can be any polynomial. Further, when we say an approximation is good it is implicit that we mean that it is good with very high probability (greater than $(1 - \frac{1}{2^{cn}})$ for some $c > 0$).

We define an experiment that has a random variable that is a good approximation to both $\Pr_{g \in s \diamond w}(C^g_n)$ and $\Pr_{f \in \mathcal{F}^n}(C^f_n)$. A direct result is that for most $w \in \mathcal{F}^n$ the value $|\Pr_{g \in s \diamond w}(C^g_n) - \Pr_{f \in \mathcal{F}^n}(C^f_n)|$ is small, and the result follows.

The major work involved in proving this lemma involves showing that the random variable in the experiment approximates both of the aforementioned values.

We define an experiment in which we draw uniformly at random a function $w \in \mathcal{F}^n$ and a set of $p(n)$ keys from $\{0,1\}^{2n}$ for the $\diamond$ operator, $\{(k_i^1 \bullet k_i^2)\}$, where p is a polynomial. We define the random variable:

$$\frac{1}{p(n)} \sum_{i=1}^{p(n)} C_n^{(s \diamond_{k_i^1 \bullet k_i^2} w)} \tag{1}$$

It is clear, by the Chernoff bound, that p can be chosen so that (1) is a good approximation of $\Pr_{g \in s \diamond w}(C_n^g)$.

In order to demonstrate that (1) also approximates $\Pr_{f \in \mathcal{F}^n}(C_n^f)$, we show that it is a good approximation of a second random variable, which itself closely approximates $\Pr_{f \in \mathcal{F}^n}(C_n^f)$.

We define a second experiment as choosing uniformly at random $q(n)$ functions from $\mathcal{F}^n$, where q is a polynomial. We define the second random variable as:

$$\frac{1}{q(n)} \sum_{i=1}^{q(n)} C_n^{f_i}. \tag{2}$$

By the Chernoff bound, for an appropriate q , the random variable (2) is a good approximation for $\Pr_{f \in \mathcal{F}^n}(C_n^f)$. Therefore, it suffices to show that the random variable (1) is a good approximation for (2).

We show that (1) and (2) are good approximations of each other by defining a third experiment in which both random variables can be calculated. In this experiment, with very high probability the random variables are equal, and therefore they are good approximations of each other.

In the third experiment we draw uniformly at random a polynomial (in n) number of random strings, $\{r_i\}$, from $\{0,1\}^n$ and a polynomial number of keys from $\{0,1\}^{2n}$ for the $\diamond$ operator, $\{k_i^1 \bullet k_i^2\}$.

Observe that the random variable (2) can be calculated in this experiment: any call to an oracle-gate during the computation of C^{f_i} can be answered with a random bit-string r_j. (Recall C is of a special form: it never makes the same oracle query twice.)

Unfortunately, it's not as easy to calculate (1) in the third experiment. As w was chosen at random in the first experiment, for any i we can calculate the value $C^{(s \diamond_{k_i^1 \bullet k_i^2} w)}$ by replacing the outputs of the oracle gates with random bit-strings. Unfortunately, the calculation of (1) requires the evaluation of $C^{(s \diamond_{k_i^1 \bullet k_i^2} w)}$ for a polynomial number of values of i. These evaluations are not independent, and therefore the scheme used to calculate (2) is not a valid method for computing (1). The problem is that during the evaluations of $C^{(s \diamond_{k_a^1 \bullet k_a^2} w)}$ and $C^{(s \diamond_{k_b^1 \bullet k_b^2} w)}$ the respective queries x and y could be made to oracle gates, where $x \oplus k_a^2 = y \oplus k_b^2$, and in such cases the outputs of the gates are dependent on each other. Fortunately, we can show that the probability of such an event occurring is

negligible and that this is the only case in which we cannot replace the output of the oracle gates with random strings to simulate the calculation of (1). Therefore, with high probability, the values of (1) and (2) are equal in third experiment. The lemma follows. □

3.3 Proof of the Isolation Lemma

Assume that there exists a polynomial-sized decision-circuit family $\{C_n\}$ which for some constant $c > 0$ and infinitely many n, $\left|\Pr_{g\in\mathsf{G}^n\diamond\mathsf{H}^n}(C_n^g) - \Pr_{f\in\mathcal{F}^n}(C_n^f)\right| \geq \epsilon(n)\delta(n)+\frac{1}{n^c}$. WLOG we assume that $\Pr_{g\in\mathsf{G}^n\diamond\mathsf{H}^n}(C_n^g)-\Pr_{f\in\mathcal{F}^n}(C_n^f) \geq \epsilon(n)\delta(n)+\frac{1}{n^c}$, as otherwise we can simply flip the output bit of C_n.

Lemma 6. *For $i > 0$ and for each n let*

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

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

Proof. Suppose for contradiction that there exists an i and j such that for infinitely many n $\Pr_{w\in\mathcal{F}^n}(w \in S^n(i)) \geq \frac{1}{n^j}$. We will show this contradicts Lemma 5. We first note that since $\Pr_{\phi\in\mathsf{G}^n\diamond S^n(i)}(C_n^\phi) \geq \Pr_{f\in\mathcal{F}^n}(C_n^f) + \frac{1}{n^i}$, then by an averaging argument we can fix a $g \in \mathsf{G}^n$ such that $\Pr_{h\in g\diamond S^n(i)}(C_n^h) \geq \Pr_{f\in\mathcal{F}^n}(C_n^f) + \frac{1}{n^i}$. Then using the first moment method we note that given g, there must be a fraction $\frac{1}{n^{2i}}$ of $w \in S^n(i)$ which have the "good" property that $\Pr_{\psi\in g\diamond w}(C_n^\psi) \geq \Pr_{f\in\mathcal{F}^n}(C_n^f) + \frac{1}{n^{2i}}$. Since $S^n(i)$ is also a "significant" $(\frac{1}{n^j})$-fraction of $\mathcal{F}^n$, the probability that a random w has the "good" property is $\frac{1}{n^{2i+j}}$, and this contradicts Lemma 5. □

Lemma 7. *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:*

$$\left| \Pr_{g\in\mathsf{G}^n}(\Xi_n^g) - \Pr_{f\in\mathcal{F}^n}(\Xi_n^f) \right| \geq \delta(n) + \frac{1}{n^{2c}};$$

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

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

Proof. Suppose it is the case that for infinitely many n there exists an $h^n \in \mathsf{H}^n$ such that $\left|\Pr_{g\in\mathsf{G}\diamond h^n}(C_n^g) - \Pr_{f\in\mathcal{F}^n}(C_n^f)\right| \geq \delta(n)+\frac{1}{n^{2c}}$. For each such n we create a decision circuit Ξ_n, where $\Xi_n^w = C_n^{(w\diamond h^n)}$. We observe that:

$$\begin{aligned}
|\Pr_{\psi\in\mathsf{G}^n}(\Xi_n^\psi) - \Pr_{f\in\mathcal{F}^n}(\Xi_n^f)| &= |\Pr_{\psi\in\mathsf{G}^n\diamond h^n}(C_n^\psi) - \Pr_{f\in\mathcal{F}^n\diamond h^n}(C_n^f)| \\
&= |\Pr_{\psi\in\mathsf{G}^n\diamond h^n}(C_n^\psi) - \Pr_{f\in\mathcal{F}^n}(C_n^f)| \qquad \text{(Corollary 1)} \\
&\geq \delta(n) + \frac{1}{n^{2c}}
\end{aligned}$$

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 $7n \leq C_{\mathsf{H}}(n)$, giving us a circuit of size $\leq s_C(n) + Q_{C_n}(2C_{\mathsf{H}}(n))$. □

Main Argument We now present the main argument for proving the Diamond Isolation Lemma. WLOG, we assume that

$$\Pr_{g \in \mathsf{G}^n \diamond \mathsf{H}^n}(C_n^g) - \Pr_{f \in \mathcal{F}^n}(C_n^f) \geq \epsilon(n)\delta(n) + \frac{1}{n^c}, \tag{3}$$

if this is not the case flip the output bit of C_n.

We assume that there exists no family of circuits $\{\Xi_n\}$, where each circuit Ξ_n is of size $c_{\mathsf{H}}(n) + s_C(n)$, such that for infinitely many n:

$$\left| \Pr_{g \in \mathsf{H}^n}(\Xi_n^g) - \Pr_{f \in \mathcal{F}^n}(\Xi_n^f) \right| \geq \delta(n) + \frac{1}{n^{2c}}.$$

From the above assumption and Lemma 7, we know that for all sufficiently large n and all $h^n \in \mathsf{H}^n$:

$$\left| \Pr_{\psi \in \mathsf{G} \diamond h^n}(C_n^\psi) - \Pr_{f \in \mathcal{F}^n}(C_n^f) \right| < \delta(n) + \frac{1}{n^{2c}}. \tag{4}$$

We now outline the argument. By (3), C_n accepts a fraction of $\mathsf{G}^n \diamond \mathsf{H}^n$ which is "significantly larger" than $\epsilon(n)\delta(n) + \Pr_{C_n}(\mathcal{F}^n)$. However, by (4), for each $h \in \mathsf{H}^n$ not much more than a $\delta(n) + \Pr_{C_n}(\mathcal{F}^n)$ fraction of the functions in $\mathsf{G}^n \diamond h$ are accepted by C_n. As $\Pr_{\phi \in \mathsf{G} \diamond \mathsf{H}}(C_n^\phi)$ is the expected value of $\Pr_{\phi \in \mathsf{G} \diamond h}(C_n^\phi)$ over the distribution H^n, it must be the case that $\Pr_{\phi \in \mathsf{G} \diamond h}(C_n^\phi)$ is "significantly larger" than $\Pr_{f \in \mathcal{F}^n}(C_n^f)$ for at least an $\epsilon(n)$ fraction of the $h \in \mathsf{H}^n$. Given a function ω our distinguishing circuit will approximate $\Pr_{\psi \in \mathsf{G} \diamond \omega}(C_n^\psi)$ and accept if it is "significantly larger" than $\Pr_{f \in \mathcal{F}^n}(C_n^f)$. By the above argument this will accept an $\epsilon(n)$ fraction of the functions in H^n and, by Lemma 6, the same circuit will accept almost no random functions in $\mathcal{F}^n$. We now give the details of the proof outlined above.

Since we cannot compute $\Pr_{\phi \in \mathsf{G}^n \diamond \omega}(C_n^\phi)$ in polynomial time, we approximate it with the probabilistic circuit A_n:

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

where $g_1, ..., g_{n^b} \in \mathsf{G}^n$ and $k_1^1, k_1^2, .., k_{n^b}^1, k_{n^b}^2 \in \{0,1\}^n$ 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 such that:

$$\Pr_{w \in \mathcal{F}^n}\left[\left| A_n^w - \Pr_{\phi \in \mathsf{G}^n \diamond w}(C_n^\phi) \right| \geq \frac{1}{n^{3c}} \right] \leq \frac{1}{2^{n^{2\alpha}}},$$

and

$$\Pr_{h\in\mathsf{H}^n}\left[\left|A_n^h - \Pr_{\phi\in\mathsf{G}^n\diamond h^n}(C_n^\phi)\right| \geq \frac{1}{n^{3c}}\right] \leq \frac{1}{2^{n^{2\alpha}}}.$$

Since we want a deterministic circuit we derandomize A_n, by Lemma 2, to get the circuit B_n, such that for all but $\frac{1}{2^{n^\alpha}}$ of the $w \in \mathcal{F}^n$:

$$\left|B_n^w - \Pr_{\phi\in\mathsf{G}^n\diamond w}(C_n^\phi)\right| < \frac{1}{n^{3c}}, \tag{5}$$

and for all of the $h \in \mathsf{H}^n$:

$$\left|B_n^h - \Pr_{\phi\in\mathsf{G}^n\diamond h}(C_n^\phi)\right| < \frac{1}{n^{3c}}, \tag{6}$$

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

Claim.

$$\Pr_{h\in\mathsf{H}^n}[B_n^h \geq \Pr_{f\in\mathcal{F}^n}(C_n^f) + \frac{1}{n^{2c}}] \geq \epsilon(n) + \frac{1}{n^{2c}}.$$

Proof. Assume for contradiction that $\Pr_{h\in\mathsf{H}^n}[B_n^h \geq \Pr_{f\in\mathcal{F}^n}(C_n^f) + \frac{1}{n^{2c}}] < \epsilon(n) + \frac{1}{n^{2c}}$. Let $\mathcal{K}^n \subseteq \mathsf{H}^n$ be the set of functions $h \in \mathsf{H}^n$, for which $B_n^h \geq \Pr_{f\in\mathcal{F}^n}(C_n^f) + \frac{1}{n^{2c}}$, and let $\overline{\mathcal{K}^n}$ be its complement.

$$\begin{aligned}
\Pr_{\phi\in\mathsf{G}\diamond\mathsf{H}}(C_n^\phi) - \Pr_{f\in\mathcal{F}^n}(C_n^f) &= \sum_{h\in\mathcal{K}^n}\left(\left(\Pr_{\phi\in\mathsf{G}\diamond h}(C_n^\phi) - \Pr_{f\in\mathcal{F}^n}(C_n^f)\right)\Pr_{\psi\in\mathsf{H}^n}[\psi = g]\right) + \\
&\quad \sum_{h\in\overline{\mathcal{K}^n}}\left(\left(\Pr_{\phi\in\mathsf{G}\diamond h}(C_n^\phi) - \Pr_{f\in\mathcal{F}^n}(C_n^f)\right)\Pr_{\psi\in\mathsf{H}^n}[\psi = h]\right) \\
&\leq \sum_{h\in\mathcal{K}^n}\left(\left(\Pr_{\phi\in\mathsf{G}\diamond h}(C_n^\phi) - \Pr_{f\in\mathcal{F}^n}(C_n^f)\right)\Pr_{\psi\in\mathsf{H}^n}[\psi = h]\right) + \\
&\quad \sum_{h\in\overline{\mathcal{K}^n}}\left(\left(\left(B_n^h - \Pr_{f\in\mathcal{F}^n}(C_n^f)\right) + \frac{1}{n^{2c}}\right)\Pr_{\psi\in\mathsf{H}^n}[\psi = h]\right) \\
&\leq \sum_{h\in\mathcal{K}^n}\left(\left(\Pr_{\phi\in\mathsf{G}\diamond h}(C_n^\phi) - \Pr_{f\in\mathcal{F}^n}(C_n^f)\right)\Pr_{\psi\in\mathsf{H}^n}[\psi = h]\right) + \\
&\quad \left(1 - \epsilon(n) - \frac{1}{n^{2c}}\right)\frac{1}{n^{2c}} \qquad (7) \\
&\leq \left(\epsilon(n) + \frac{1}{n^{2c}}\right)\left(\delta(n) + \frac{1}{n^{2c}}\right) + \frac{1}{n^{2c}} \qquad (8) \\
&< \epsilon(n)\delta(n) + \frac{1}{n^c}. \qquad \text{(contradiction)} \qquad (9)
\end{aligned}$$

Equation (7) follows from two facts. First, by assumption, the probability that a random $h \in \mathsf{H}^n$ is in $\overline{\mathcal{K}^n}$ is $1 - \epsilon(n) - \frac{1}{n^{2c}}$. Second, for each $h \in \overline{\mathcal{K}^n}$, $B_n^h -$

$\Pr_{f\in\mathcal{F}^n}(C_n^f) < \frac{1}{n^{2c}}$. Equation (8) follows from two facts. First, by assumption, $\Pr_{h\in\mathsf{H}^n}[h \in \mathcal{K}^n] < \epsilon(n) + \frac{1}{n^{2c}}$. Second, by (4), for each $h \in \mathsf{H}^n$, $\Pr_{\phi\in\mathsf{G}\diamond h}(C_n^\phi) - \Pr_{f\in\mathcal{F}^n}(C_n^f) < \delta(n) + \frac{1}{n^{2c}}$. Equation (9) contradicts the fact that $\Pr_{\phi\in\mathsf{G}\diamond\mathsf{H}}(C_n^\phi) - \Pr_{f\in\mathcal{F}^n}(C_n^f) \geq \epsilon(n)\delta(n) + \frac{1}{n^c}$. □

We create the decision circuit Υ_n^w which accepts w iff $B_n^w \geq \Pr_{f\in\mathcal{F}^n}(C_n^f) + \frac{1}{n^{2c}}$.

$$\begin{aligned}
\Pr_{h\in\mathsf{H}^n}(\Upsilon_n^h) - \Pr_{f\in\mathcal{F}^n}(\Upsilon_n^f) &\geq \epsilon(n) + \frac{1}{n^{2c}} - \Pr_{f\in\mathcal{F}^n}(\Upsilon_n^f) \\
&\geq \epsilon(n) + \frac{1}{n^{2c}} - \frac{1}{2^{n^\alpha}} - \\
&\quad \Pr_{w\in\mathcal{F}^n}\left[\Pr_{g\in\mathsf{G}^n\diamond w}(C_n^g) \geq \Pr_{f\in\mathcal{F}^n}(C_n^f) + \frac{1}{n^{2c}} - \frac{1}{n^{3c}}\right] \quad (10) \\
&\geq \epsilon(n) + \frac{1}{n^{2c}} - \frac{1}{n^{3c}} - \frac{1}{2^{n^\alpha}} \quad (11) \\
&\geq \epsilon(n) + \frac{1}{n^{3c}} \qquad \text{(For sufficiently large } n\text{)}
\end{aligned}$$

Equation (10) follows as B_n^ω approximates $\Pr_{g\in\mathsf{G}^n\diamond\omega}(C_n^g)$ to within a factor of $\frac{1}{n^{3c}}$ for all but $\frac{1}{2^{n^\alpha}}$ of the $\omega \in \mathcal{F}^n$. Equation (11) follows by a direct application of Lemma 6.

By performing the straightforward construction of Υ_n, we see that there does exist a fixed polynomial p mentioned in the statement of the lemma for which the size of Υ_n is bound by $p(n^c \cdot c_{\mathsf{G}}(n))s_C(n)$. □

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, as described in the introduction, 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 – possibly by the time proven method of using composition.

Further, as one of the anonymous referees pointed out, it appears possible to show in the Kilian Rogaway model [6] that the construction can be used to increase the effective key-length of a block-cipher. This would appear to give further evidence of the benefit of using the construction in practice. Further, since the construction is parallelizable it may be preferable to 3-DES for extending the key-lengths of DES. However, since the resulting generator is a function generator and not a permutation generator, there will be systems and applications where this is an infeasible approach.

5 Acknowledgments

The author would like to thank Charles Rackoff for suggesting the problem and for many valuable discussions and suggestions.

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, 1998.
2. O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986.
3. O. Goldreich, N. Nisan, and A. Wigderson. On yao's xor-lemma. `http://theory.lcs.mit.edu/~oded/`, 1995.
4. J. Hastad, R. Impagliazzo, L.A. Levin, and M. Luby. Construction of pseudo-random generator from any one-way function. *Accepted to the SIAM Journal of Computing*, 28(4):1364–1396, 1998.
5. R. Impagliazzo. Hard core distributions for somewhat hard problems. `http://www-cse.ucsd.edu/~russell/`, 1994.
6. 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, 1996.
7. L.A. Levin. One-way functions and pseudorandom generators. *Combinatorica*, 7(4):357–363, 1987.
8. M. Luby and C. Rackoff. Pseudo-random permutation generators and cryptographic composition. In *Proceedings of the 18th Annual Symposium on Theory of Computing*, pages 353–363. ACM, 1986.
9. M. Luby and C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal on Computing*, 17:373–386, 1988.
10. Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.
11. Steven Myers. *On the Development of Pseudo-Random Function Generators and Block-Ciphers using the XOR and Composition Operators*. M.Sc. Thesis. University of Toronto, Canada, 1999.
12. Moni Naor and Omer Reingold. On the construction of pseudo-random permutations: Luby-Rackoff revisited. *Journal of Cryptology*, 12(1):29–66, 1999.
13. Andrew Yao. Theory and applications of trapdoor functions (extended abstract). In *Proceedings of the 23rd Symposium on Foundations of Computer Science*, pages 80–91. IEEE, 1982.