On the Development of Block-Ciphers and Pseudo-Random Function Generators Using the Composition and XOR Operators

by

Steven A. Myers

A thesis submitted in conformity with the requirements
for the degree of Master's Of Science
Graduate Department of Computer Science
University of Toronto

Abstract

On the Development of Block-Ciphers and Pseudo-Random Function Generators Using the Composition and XOR Operators

Steven A. Myers

Master's Of Science

Graduate Department of Computer Science

University of Toronto

1999

We attempt to provide evidence for the security of block-ciphers which are constructed by taking the composition and exclusive-or of many non-secure function generators. We attempt to provide this evidence by showing that such construction can be used to combine partially secure pseudo-random function and permutation generators into generators with stronger security properties than any of their constituents. We extend results of Luby and Rackoff, and show that there are constructions based on the composition and $\Box$ operators which can be used to combine $1-\delta$ secure pseudo-random permutation and function generators, where $0 < \delta < 1$, to achieve a $1 - \frac{1}{\log^c n}$ secure generator, for any $c > 0$. We then give the first proof that there is a natural construction which combines together $1-\delta$ secure function generators to form a pseudo-random function generator. The last construction leads us to question whether or not security increasing constructions in the pseudo-random model actually provide evidence that such constructions are useful in the design of block ciphers.

Acknowledgements

No large work is completed by a sole person, and therefore I owe my thanks and appreciation to many people for their help and encouragement in completing this work. Foremost, I would like to thank Charlie Rackoff for both introducing me to the field of cryptography and sharing his extensive knowledge of the field with me. Charlie's ability to see to the heart of any problem has been inspirational, and his willingness to discuss problems and provide feedback have been invaluable at all stages of my research and thesis preparation.

I would like to thank Allan Borodin for reading a draft of this thesis and suggesting some important changes and corrections which improved the final product.

I would like to thank my family for their caring and support throughout my life. They have provided me with opportunities to pursue my interests and encouragement to achieve my goals.

I would like to thank Catherine Ponting for providing me with moral support and encouragement, throughout the writing of this thesis.

Finally, I would like to thank the Department of Computer Science and the University of Toronto for their scholarships and financial support.

Contents

Chapter 1

Introduction

The field of cryptography has a long and rich history, yet it is only in the past few decades that work in the field has become more of a science then an art. Previously, cryptographic systems were developed based on heuristics, and the cleverness of their designers. However, the designers gave no evidence that the techniques or operators they used in the development of their systems were actually increasing security.

Two operators which have consistently been used in the development of block ciphers are the composition and exclusive-or (XOR) operators, for it seems to have been taken for granted in the cryptographic community that these operators increase security. This belief seems to stem from empirical evidence that cryptographic systems based on these operators have remained fairly secure, under the scrutiny of very intense and extensive cryptanalysis. One such example is the Data Encryption Standard (DES), in which 16 permutations generators, which are "completely insecure" individually, are composed and combined together with the composition and XOR operators, to give what appears to be a remarkably secure block cipher.

Since the development of DES, in the early 1970s, it has undergone extensive public cryptanalysis, and during the past 30 years only two forms of cryptanalysis have suggested attacks which require significantly fewer than the 2^{55} calls to the DES algorithm which are necessary to break it by a brute-force attack on its 56 bit key. The two forms of cryptanalysis which have performed reasonably well are linear, and differential cryptanalysis developed by M. Matsui in [16]; and E. Biham and A. Shamir in [3], respectively. Using the techniques of linear cryptanalysis, 2^{47} executions of DES on *known* plain texts are required to break it; where as in differential cryptanalysis, 2^{47} executions on *chosen* plain texts are required. In either case, the attacks are not significantly stronger than a brute force attack on the keys of DES, and the empirical evidence seems to suggest that DES is a very secure cryptographic system.

Further evidence of the pervasiveness of the use of composition and XOR as operators for augmenting the security of block ciphers is the fact that MARS, TWOFISH, RC6 and most of the other Advanced Encryption Standard (AES) candidate ciphers seem to rely on these operators to increase their security. All of the candidate ciphers are currently in a competition to become the Advanced Encryption Standard. The AES will replace DES as the standard block cipher used by the U.S. Government, and therefore the candidate algorithms are expected to be very secure, given the current state of knowledge in cryptology.

The purpose of this thesis is to provide evidence which supports the use of composition and XOR operators in the development of block ciphers. We will not be able to demonstrate that their use in the development of a cryptographic system can result in a provably secure cryptographic system, as such a proof would imply $\mathcal{P} \neq \mathcal{NP}$, and thus solve one of the most important open questions in computer science. Instead, we have much more modest goals, and will demonstrate that there are constructions based on composition and XOR which combine function and permutation generators which are already partially secure, and which result in function or permutation generators which are more secure. This will be used to give evidence to suggest that composition improves security, and partially justifies their use in the design of block ciphers.

1.1 Overview of the Thesis

We will now provide a brief outline of the thesis. We begin, in Chapter 2, by introducing some standard definitions of cryptographic primitives. We also explain the relationships that exist between these primitives. Then in Chapter 3 we look at some previous work which has been done on constructions which increase the security of different types of cryptographic primitives. We will look at work which has been done in both complexity and information theoretic models.

In Chapter 4, we will then develop a list of possible constructions which use composition and XOR operators to combine partially secure function generators into another function or permutation generator. In the model we will work in, our definition of security will be based on complexity theoretic notions. We will show that most of our proposed constructions result in a function or permutation generator which is at best as secure as its constituent generators. Therefore these constructions are not useful for increasing security in the construction of block ciphers, but based on some work done by Luby and Rackoff in [12], in Chapters 5, 6 and 7 we will prove that three of the constructions result in generators which have stronger security than any of their constituent generators.

The first security increasing construction is based on the composition operator. Previously in [12], Luby and Rackoff have shown that composing partially secure permutation generators a constant number of times results in generator with stronger security properties than any of the constituents. By making some modifications to their argument, by making it less dependent on sampling different distributions of functions, we will extend their result to show that we can compose a $\mathcal{O}(\log\log n)$ number of partially secure permutation generators. This will result in a construction which is more secure than the constructions which are provably secure from the initial result by Luby and Rackoff. These results are presented in Chapter 5.

The second security increasing construction is based on the $\Box$ operator, proposed by Luby and Rackoff in [13], which is a combination of the composition and XOR operators.

Given two functions f_1 and f_2 such that $f_i : \{0,1\}^n \to \{0,1\}^n$, for $i \in \{1,2\}$, define the operator $\Box$ (read as box) as:

$$(f_1 \Box f_2)(x) = f_1(f_1(x) \oplus f_2(x)) \oplus f_2(f_1(x) \oplus f_2(x)).$$

By proving two technical arguments about this operator, we will show that we can use the arguments previously presented for the composition operator acting on permutation generators, and apply them to the case of the $\Box$ operator acting on function generators. Therefore, it will follow that we can combine $\mathcal{O}(\log\log n)$ function generators using the $\Box$ operator, and achieve the same security as was achieved by composing together permutation generators. These results are presented in Chapter 6.

The third security increasing construction is based on the $\Diamond$ operator generator. We define the $\Diamond$ operator generator (read as Diamond) as $\Diamond = \{\Diamond^n_{r_1,r_2} | n \in \mathbb{N} \wedge r_1, r_2 \in \{0,1\}^n\}$. 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 as:

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

We will show that by combining function generators using randomly chosen operators from the $\Diamond$ operator generator, we can combine any polynomial number of function generators, and the result will be a completely secure[1] function generator. This proof will be based on the proof by Luby and Rackoff, but we will observe that this construction allows us to short-circuit a key section of the proof. This in turn is what allows us to combine together many more function generators than in the previous cases. We will then observe that while the third construction improves security significantly, there is no reason to suspect that it will be useful in the design of block ciphers. These results are presented in Chapter 7.

Finally, in Chapter 8 we will suggest future research directions and goals which are

[1] The notion of completely secure will formally be define in the sequel

similar but more ambitious than those presented in this document. In particular, we suggest the development of a theory for the development of block-ciphers. In this theory we would conjecture that completely insecure function generators could be combined in different constructions –such as the ones mentioned above– to form secure generators. We would say that a construction was secure if the initial, insecure generators, and the constructed generator had certain combinatorial properties. The motivation for such a theory is that it would allow for the comparison of different resource and design trade-offs which are made during the development of different ciphers.

1.2 Notation & Terminology

Below we introduce some notation and terminology which will be used throughout the rest of the thesis.

Terminology 1.1 *For $\mu, \nu \in \{0,1\}^*$, let $\mu \bullet \nu$ denote their concatenation.*

Terminology 1.2 *Let $\mathcal{P}$ denote the set of all permutations, and let $\mathcal{P}^n$ denote the set of all permutations $\sigma : \{0,1\}^n \to \{0,1\}^n$.*

Terminology 1.3 *Let $\mathcal{F}$ denote the set off all functions, $\mathcal{F}^{l \to p}$ denote the set of all functions $f : \{0,1\}^l \to \{0,1\}^p$, and let $\mathcal{F}^n$ be the set $\mathcal{F}^{n \to n}$.*

Terminology 1.4 *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 $f(\alpha) \oplus g(\alpha)$.*

Terminology 1.5 *For any two functions g and h, let $g \circ h(x) = g(h(x))$. When we refer to the composition $f \circ g(x) = f(g(x))$ of two functions $f(x)$ and $g(x)$, we refer to f as the* outer *function, and g as the* inner *function.*

Terminology 1.6 *For any set A, let $x \in A$ be the action of uniformly at random choosing an element x from A. It will be clear from context when $\in$ is used to refer to an element in a set, and when it refers to choosing uniformly from a set.*

Terminology 1.7 *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, $\Pr_{\mathcal{D}_n} e_n < \frac{1}{n^c}$.*

Terminology 1.8 *When we refer to a function poly(n) we are referring to any function in the set $\bigcup_{i=1}^{\infty} \mathcal{O}(n^i)$.*

Terminology 1.9 *When we refer to a function poly-log(n) we are referring to any function in the set $\bigcup_{i=1}^{\infty} \mathcal{O}(\log^i n)$.*

Terminology 1.10 *Let A be an algorithm or Turing Machine, and let π be an orcale. We denote by A^π the algorithm or Turing Machine A, with access to the oracle π.*

Terminology 1.11 *We call a circuit C a* probabilistic circuit, *if it requires as inputs a string of random bits.*

Terminology 1.12 *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 $C(x) = 0$.*

Terminology 1.13 *Let $\mathcal{D}$ be a distribution over the inputs of a decision circuit C. Then we define*

$$\Pr_C(\mathcal{D}) = \Pr_{d \in \mathcal{D}}[C(d) = 1],$$

and we say that C accepts *a* fraction $\Pr_C(\mathcal{D})$ *of its inputs, and* rejects *a* fraction $1 - \Pr_C(\mathcal{D})$ *of its inputs.*

Terminology 1.14 *Let $\alpha \in \{0, ..., 2^n - 1\}$, then let $\bar{\alpha}$ be the bit-wise binary representation of α in $\{0, 1\}^n$.*

Terminology 1.15 *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* $b < d$*.*

Terminology 1.16 *We call* $G : \{0,1\}^\kappa \times \{0,1\}^n \rightarrow \{0,1\}^m$ *a function generator instance. 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* $\mathcal{N} \subset \mathbb{N}$ *be an infinitely large set. For each* $n \in \mathcal{N}$*, let* $G^n : \{0,1\}^{\ell(n)} \times \{0,1\}^n \rightarrow \{0,1\}^{m(n)}$ *be a function generator. We call* $G = \{G^n | n \in \mathcal{N}\}$ *a function generator.*

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

Chapter 2

Cryptographic Primitives

Cryptographers have developed a series of conjectured primitives which can be used in the development of cryptographic protocols. Due to a lack of progress in complexity theory, no one has been able to prove the existence of these primitives. In fact, as will be shown in the sequel, such a proof would imply that $\mathcal{P} \neq \mathcal{NP}$, and would close the largest open question in computer science. In this chapter we present the definitions for a number of cryptographic primitives and some theorems which represent the relationships between them. However, in the definition of every cryptographic primitive there is the notion of an adversary. Therefore, we begin the chapter with a discussion of the two standard adversarial models, and the relationships that exist between them.

2.1 Uniform vs. Non-Uniform Adversaries

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, were they trying to break existential instances of primitives. There are two standard computational models which are used to define resource bounded adversaries.

Uniform Adversaries

A uniform adversary is a Turing Machine which runs in time bounded by a polynomial in the size of its input. In practice we know of algorithms which run more quickly using randomization then the best known non-randomized versions, so we allow the Turing machine to be probabilistic and have access to an infinite string of random bits. In order to model the adversaries of certain primitives, we allow that the Turing Machines have access to one or more oracles, which answer queries. Unless otherwise stated we assume that the oracles respond to a query in unit time.

Non-Uniform Adversary

A non-uniform adversary is an infinite set of circuits $\{C_1, C_2, ...\}$, where circuit C_i is used on inputs of size i. We wish to model efficient computation on 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.

Unlike the uniform adversary we do not need to allow our circuits to be probabilistic. Given a probabilistic circuit which takes a string of random bits as part of its input, we can use the non-uniformity of the circuits to fix a specific set of bits as the random-input of the circuit. Normally, a simple averaging argument will show that such a string exists. We refer the interested reader to [7] for a further discussion of this topic.

In correspondence with the oracles in the uniform model, we assume that circuits can have oracle-gates, which respond to inputs in the same manner an oracle would. We assume that the gates are of unit size, and are otherwise treated like any other gate. 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.

Terminology 2.1 *We write $C(f)$ to represent a circuit C which has access to the oracle f.*

Notice that in an abuse of notation, circuit inputs and circuit oracles are represented in the same manner. This is done because in this thesis, in most cases of interest, our circuits will not have inputs in the standard sense. Rather, they will be constructed to differentiate between different oracles, and therefore, in some sense, we are considering the oracle as an input. Notice that when we treat oracles as inputs to circuits we consider the size of the oracle query to be the size of input of the circuit. We hope that this abuse of notation will not cause confusion, and that we have made it clear from context whether or not the input to a circuit is an oracle or a traditional input bit string.

Terminology 2.2 *Let $\mathcal{D}$ be a distribution over oracles of an oracle-decision-circuit C. Then we define*

$$\Pr_C(\mathcal{D}) = \Pr_{d \in \mathcal{D}}[C(d) = 1],$$

and we say that C accepts *a* fraction $\Pr_C(\mathcal{D})$ *of its inputs, and* rejects *a* fraction $1 - \Pr_C(\mathcal{D})$ *of its inputs.*

In the sequel we present the primitives in the uniform model, but note that it is easy to determine the corresponding definitions for the non-uniform model.

2.2 Primitives

The first primitive we consider is the *pseudo-random number generator*, which takes a random sequence of bits and extends them into a much longer sequence of random looking bits.

Definition 2.1 (Pseudo-Random Number Generator – PRNG) *Let p be a polynomial where $p(n) > n$ for every n. For each n let $G^n : \{0,1\}^n \to \{0,1\}^{p(n)}$ be a function computable in a time bounded by a polynomial in n. Then $G = \{G^n | n \in \mathbb{N}\}$ is a* pseudo-random number generator *if, for all adversaries A, running in time bounded by a polynomial in n, and for all polynomials q, and for all sufficiently large n:*

$$\left| \Pr_{X\in\{0,1\}^n}[A(G^n(X))=1] - \Pr_{Y\in\{0,1\}^{p(n)}}[A(Y)=1] \right| < \frac{1}{q(n)}.$$

Theorem 2.1 *If there exists a PRNG G, then $\mathcal{P} \neq \mathcal{NP}$.*

Proof: Let $L = \{y | \exists n,\ \exists x \in \{0,1\}^n \text{ s.t. } G^n(x) = y\}$. Observe, there exists a non-deterministic machine M which decides L; given an input y, of size $p(n)$, M guesses a string $x \in \{0,1\}^n$ and accepts if $G^n(x) = y$. However, by the definition of G, there exists no polynomial time bounded machine which decides L, and therefore $\mathcal{P} \neq \mathcal{NP}$. ...□

An other natural primitive to consider is the *pseudo-random function generator*, which given a random string of bits generates a random looking function f.

Definition 2.2 (Pseudo-Random Function Generator – PRFG) *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 instance*, computable in time bounded by a polynomial in n. Define $G = \{G^n | n \in \mathbb{N}\}$ to be a function generator. For $k \in \{0,1\}^{\ell(n)}$, let $G^n(k,\cdot)$ be written as $g_k^n(\cdot)$ or $g_k(\cdot)$ when the value of n is clear, and referred to as the function g_k^n or g_k respectively. If A^h is an adversary which queries a function h, then we say it is capable of ϵ* distinguishing g from random *if, for $\epsilon : \mathbb{N} \to [0,1]$, some polynomial p and infinitely many n:*

$$\left| \Pr_{k\in\{0,1\}^n}\left[A^{g_k^n}=1\right] - \Pr_{f\in\mathcal{F}^{\ell(n)\to m(n)}}\left[A^f=1\right] \right| \geq \epsilon(n) + \frac{1}{p(n)}.$$

We say that G is $(1-\epsilon(n))$ secure if there exists no function querying adversary A, bound to run in time polynomial in n, which can ϵ distinguish G from random.

We say that G is a pseudo-random function generator *if it is 1 secure.*

If G is a $(1-\epsilon(n))$ secure PRFG, and if it is the case that for all n, and for all $k \in \{0,1\}^{\ell(n)}$ that g_k^n is a permutation, then we call g a $(1-\epsilon(n))$ secure *pseudo-random permutation generator*. We say that G is a **pseudo-random permutation generator (PRPG)**, if G is 1-secure.

Terminology 2.3 *If G is a 1-secure generator, we say it is completely secure. If G is a 0-secure generator, we say it is insecure, and if G is neither completely secure nor insecure, then we say that it is partially secure.*

It was shown by Goldreich *et. al.*, in [5], that the existence of a PRNG is a necessary and sufficient condition to ensure the existence of a PRFG.

Theorem 2.2 *PRNGs exist iff PRFGs exist.*

We will demonstrate the constructions, and refer the reader to the original paper for the proof of correctness.

Construction: To construct the PRFG, we consider a PRNG $G = \{g^n : \{0,1\}^n \to \{0,1\}^{2n} | n \in \mathbb{N}\}$. For any $x \in \{0,1\}^n$ we will consider $g^n(x)$ as $g(x)_0^n \bullet g(x)_1^n$, where $|g(x)_0| = |g(x)_1| = n$. We will construct a PRFG, $F = \{F^n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n | n \in \mathbb{N}\}$. For each key $x \in \{0,1\}^n$ will define the function $F^n(x, \cdot) = f_x^n$.

In order to compute $f_x^n(y)$, for $y \in \{0,1\}^n$, we define a complete binary tree of depth n. We label the root with the empty string λ, and assign it the value x. We then recursively apply the following rule to assign a label and value to all other nodes, v, in the tree. Let L_{p_v} and V_{p_v} be the respective label and value of v's parent.

- If v is a left child set $i = 0$, otherwise set $i = 1$.
- Let the label for node v be $i \bullet L_{p_v}$.
- Let the value for node v be $g^n(V_{p_v})_i$.

We define the value of $f_x^n(y)$ to be equal to the value of the leaf node labeled y.

The construction in the other direction is much simpler. Let $H = \{H^n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n | n \in \mathbb{N}\}$ be a PRFG. We construct a PRNG $G = \{g^n : \{0,1\}^n \to \{0,1\}^{2n} | n \in \mathbb{N}\}$ as follows. For each n and each $x \in \{0,1\}^n$ let $g^n(x) = H_x^n(\bar{0}) \bullet H_x^n(\bar{1})$. ...$\square$

It was shown by Luby and Rackoff in [14] that the existence of PRFGs is a necessary and sufficient condition to ensure the existence of PRPGs. Their construction is based on a generalization of DES which uses independent keys in each of the sixteen rounds of the DES protocol, rather than using a key scheduling algorithm to generate a new key in each round. This generalization of DES is commonly referred to as MDES.

Additionally, Luby and Rackoff demonstrate how to construct a PRPG which is secure against an adversary which not only has access to an oracle which computes a permutation σ, but which also has access to an oracle which computes the inverse of the permutation σ^{-1}. We call a generator a *super pseudo-random permutation generator (SPRPG)* if, it is considered a PRPG under a modified version of the PRPG definition, in which each adversary A^{σ}, is replaced by a stronger adversary $A^{\sigma,\sigma^{-1}}$.

Theorem 2.3 *PRFGs exist iff PRPGs exist.*

Construction: Let $F = \{F^n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n | n \in \mathbb{N}\}$ be a PRFG. We will demonstrate the Luby-Rackoff construction, which produces a PRPG $H = \{H^n : \{0,1\}^{3n} \times \{0,1\}^{2n} \to \{0,1\}^{2n} | n \in \mathbb{N}\}$. For each n we perform the following construction. For any strings $x \in \{0,1\}^{2n}$ and $k \in \{0,1\}^n$, let $L_x^n \bullet R_x^n = x$, where $|L_x^n| = |R_x^n| = n$ and let $g_k^n(x) = R_x^n \bullet (L_x^n \oplus f_k^n(R_x^n))$. We will define $h^n_{k_1 \bullet k_2 \bullet k_3}(x) = g^n_{k_1} \circ g^n_{k_2} \circ g^n_{k_3}(x)$. Further, to construct a SPRPG define $\widehat{H} = \{\widehat{H^n} : \{0,1\}^{4n} \times \{0,1\}^{2n} \to \{0,1\}^{2n} | n \in \mathbb{N}\}$, we let $\hat{h}^n_{k_1 \bullet k_2 \bullet k_3 \bullet k_4}(x) = g^n_{k_1} \circ g^n_{k_2} \circ g^n_{k_3} \circ g^n_{k_4}(x)$. ...□

Recent work by Naor and Reingold [18] has shown that these constructions can be modified to be less dependent on calls to the PRFG. This is desirable because calls to a PRFG can consume a considerable amount of resources. Their construction requires pair-wise independent permutation generators, and while these generators are not cryptographic primitives, they are used extensively in cryptography. Therefore, we will present a definition of pair-wise independent permutation generators, and then present the construction of Naor and Reingold.

Definition 2.3 (k-wise Independent Permutation Generator) *Let ℓ be a polynomial. For each n let $G^n : \{0,1\}^{\ell(n)} \times \{0,1\}^n \to \{0,1\}^n$ be a function, computable in time bounded by a polynomial in n, where for each n and each $k \in \{0,1\}^{\ell(n)}$ the function $G^n(k,\cdot)$ is a permutation. We write $G^n(k,\cdot)$ as $g^n_k(\cdot)$, and the function is referred to as the permutation g^n_k, or g_k if the value of n is clear from the context. We define $G^n = \{G^n | n \in \mathbb{N}\}$, and say that G is a k-*wise independent permutation generator*, if for each n and for all $x_1, ..., x_k \in \{0,1\}^n$, where $x_i \neq x_j$ for $i \neq j$; and for all $y_1, ..., y_k \in \{0,1\}^n$, where $y_i \neq y_j$ for $i \neq j$:*

$$\Pr_{q \in \{0,1\}^{\ell(n)}} \left[g^n_q(x_1) = y_1 \wedge \cdots \wedge g^n_q(x_k) = y_k \right] = \prod_{i=0}^{k-1} \frac{1}{2^n - i}.$$

If G is a $2-wise$ independent permutation generator then we say that it is pair-wise *independent.*

We now show how Naor and Reingold simplify the construction of a PRPG from a PRFG originally presented by Luby and Rackoff. Naor and Reingold show that if we consider the construction of the functions h^n and $\hat{h}^n$ proposed by Luby and Rackoff, then the first permutation $g^n_{k_3}$ in h^n, and the first and last permutations, $g^n_{k_1}$ and $g^n_{k_4}$, in $\hat{h}^n$ can be replaced by permutations chosen from a pair-wise independent permutation generator [1]. Thus if $P = \{P^n : \{0,1\}^{2n} \times \{0,1\}^n \to \{0,1\}^n | n \in \mathbb{N}\}$ is pair-wise independent permutation generator, then they replace the definition of $h^n_{k_1 \bullet k_2 \bullet k_3}(x) = g^n_{k_1} \circ g^n_{k_2} \circ g^n_{k_3}(x)$ with $h^n_{k_1 \bullet k_2 \bullet \kappa_3} = g^n_{k_1} \circ g^n_{k_2} \circ p^n_{\kappa_3}$, where $\kappa_3 \in \{0,1\}^{2n}$, to construct a PRPG. Similarly, they replace the definition of $\hat{h}^n_{k_1 \bullet k_2 \bullet k_3 \bullet k_4}(x) = g^n_{k_1} \circ g^n_{k_2} \circ g^n_{k_3} \circ g^n_{k_4}(x)$ with $\hat{h}^n_{\kappa_1 \bullet k_2 \bullet k_3 \bullet \kappa_4} = p^n_{\kappa_1} \circ g^n_{k_2} \circ g^n_{k_3} \circ p^{n^{-1}}_{\kappa_4}$, where $\kappa_1, \kappa_4 \in \{0,1\}^{2n}$, to construct a SPRPG.

Note that no construction was given to produce a PRFG from a PRPG, this is because any PRPG is a PRFG. This is due to the fact that in order to differentiate a random function $f \in \mathcal{F}^n$ from a random permutation $g \in \mathcal{P}^n$, it would be necessary

[1] Their results are actually slightly stronger, but we refer the reader to the original paper for the exact results

to demonstrate that f is either not *onto* or *one-to-one*. It can easily be shown, by the birthday attack argument[2], that the probability of finding a pair $(\alpha, \beta) \in \{0,1\}^{n \times 2}$ s.t. $f(\alpha) = f(\beta)$, with at most p(n) queries to f is exponentially small, and therefore no adversary, bounded to run in polynomial time, would be able to distinguish a permutation from a function with a significant probability.

We now consider a primitive which at first glance seems unrelated to the primitives which have previously been seen.

Definition 2.4 (One-Way Functions) *For each n, let $f^n : \{0,1\}^n \to \{0,1\}^{l(n)}$ be a function computable in time bounded by a polynomial in n. Define $F = \{f^n | n \in \mathbb{N}\}$. We say that an adversary is capable of $\epsilon(n)$ inverting F if for some some polynomial p, and all sufficiently large n:*

$$\left| \Pr_{X \in \{0,1\}^n} (f^n(A(f^n(x))) = f^n(x)) - \frac{1}{2^{l(n)}} \right| \geq \epsilon(n) + \frac{1}{p(n)}.$$

*We say that F is a $\delta(n)$-*weak one-way function*, if no adversary A is capable of* $1 - \delta(n)$ *inverting F. We say that F is a* one-way *function if it is* 1*-weak. Finally, if F is a permutation, then it may be referred to as a $\delta(n)$-*weak one-way permutation*, or* one-way permutation *respectively.*

Intuitively, a $\epsilon(n)$-weak one-way function is a function which is easy to compute, but which is hard to invert on a $1 - \epsilon(n)$ fraction of its range. Similarly, a one-way function is easy to compute, but hard to invert. It may seem that assuming the existence of a weak one-way function is a weaker assumption than assuming the existence of a one-way function. This however is not the case, given a weak one-way function f, it is possible to construct a one-way function g.

Theorem 2.4 *There exists a $\delta(n)$-weak one-way function iff there exits a one-way function.*

[2]See [4] for a description of the birthday attack argument.

We refer the interested reader to [7, 15] for a proof of the above theorem.

Although one-way functions appear to have little in common with the pseudo-random primitives, it can be shown that the existence of one-way functions is a necessary and sufficient condition for the existence of PRNGs.

Theorem 2.5 *There exist PRNGs* iff *there exist one-way functions.*

Although producing a one-way function from a PRNG is a relatively trivial task, producing a PRNG from a one-way function is much more difficult. We refer the interested reader to [7, 15] for a proof of the above theorem.

Chapter 3

Related Work

In this thesis we are trying to show that there are constructions based on composition and XOR which can be used to take function and permutation generators with relatively weak security properties and use them to construct function and permutation generators with strengthened security properties. In this chapter we will review some of the work which has been done which has a relation to constructions based on XOR and composition, and which increase the security of weak primitives.

3.1 Complexity Theoretic Model

We will first consider the work which is based in complexity theory. This means that the security definitions in this section will rely on the fact that our adversaries are computationally limited in performing their tasks.

3.1.1 Yao's XOR Lemma

If we are given a predicate which is weakly unpredictable in some resource bounded computational model; then a natural question to ask is whether it is possible to construct, by some "natural" method, a new predicate which amplifies the unpredictability of the original function. Ultimately, the constructed predicate should be intractable to predict, better than at random, under a resource bounded computational model which is similar –if not identical– to the one given for the weakly unpredictable predicate. In [21], Yao postulated that the answer to this question was positive and suggested a solution that is both simple and intuitive: if $p(x)$ is weakly unpredictable then surely $P(x_1, ..., x_t) = \oplus_{i=1}^{t} p(x_i)$ is almost completely unpredictable for large enough t. Yao did not present a proof, and the first proof presented was by Levin in [11]. Since then alternative proofs have appeared from Impagliazzo [8] and from Goldreich, Nisan and Wigderson [6]. In addition to giving their own proof of the lemma, Goldreich, Nisan and Wigderson [6] provide a survey of Levin and Impagliazzo's results.

Before the XOR Lemma there had been many proofs which showed that certain constructions maintained the security of their components. However, the true importance of this result is that it was the ground breaking work which shows that there exist constructions based on partially secure components, but which are strictly more secure then any of their components.

3.1.2 PRPGs and Cryptographic Composition

Cryptographers have noted that DES is effectively the composition of 16 completely insecure permutation generators. Because DES has withstood much cryptanalysis it is often both considered to be secure and conjectured to be a PRPG. This led Luby and Rackoff to define the notion of a partially secure PRPG and conjecture that the composition of several partially secure PRPGs resulted in a PRPG with stronger security then any of its components. They proved this conjecture to be true in [12]. Later we will use their notion of partially secure generators to study other constructions which appear to be security increasing. Additionally, many of the proofs in this thesis will be based on the proof that Luby and Rackoff presented in [12].

3.2 Perfect Cipher Model

Previously we have seen work where the composition and the XOR operators have been used to increase the security of partially secure cryptographic primitives. In this section we assume that *block-ciphers* (or just cipher for short) exist. A cipher is simply a function generator $F : \{0,1\}^{\kappa} \times \{0,1\}^n \to \{0,1\}^n$, which generates functions which intuitively are supposed to look random to computationally limited adversaries. Note that there is no notion of asymptotic security, as opposed to the definitions of PRFGs and PRPGs. The notion of ciphers corresponds to real world cryptographic functions such as DES, and AES which are just specific function generators, and not defined asymptotically for every n. In this section we present some work which asks the question of how secure ciphers can be combined so that the minimum number of resources needed to break the construction is more than the number of resources necessary to break the individual cipher.

3.2.1 DES Transforms & Generic Attacks

When DES was first introduced it came under criticism for its small 56-bit key length. People feared that it would be feasible to mount an attack on DES by performing a brute force search on the key space of DES. As the speed of computers has greatly increased since the release of DES, in 1975, this attack has become a serious threat, and in 1993 Weiner provided an estimate that for $1,000,000 someone could build a machine which broke DES in an average time of 3.5 hours. In 1999 the cost of such a machine has surely decreased, and the average amount of time needed to break DES is probably much smaller, given the faster processors currently available.

Cryptographers questioned if it was possible to increase the key-length of DES while maintaining the time-tested security of the DES cipher. One solution which was proposed by Rivest, in 1984, is called DESX. We define DESX as $\mathrm{DESX}_{k_1,k_2,k}(x) = k_1 \oplus \mathrm{DES}_k(k_2 \oplus x)$, where $k_1, k_2 \in \{0,1\}^{64}$ and $k \in \{0,1\}^{56}$. Of course this is easily generalized to arbitrary function and permutation generators. If $\mathrm{F} : \{0,1\}^\kappa \times \{0,1\}^n \rightarrow \{0,1\}^n$ is a function generator then define $\mathrm{FX}_{k_1,k_2,k}(x) = k_1 \oplus \mathrm{F}_k(k_2 \oplus x)$, where now $k_1, k_2 \in \{0,1\}^n$ and $k \in \{0,1\}^\kappa$.

In [10],[20] and [9] Kilian and Rogaway show that FX does in fact increase the effective key length of a cipher F. However, it is not easy to define a model in which brute-force key-searches on a cipher and a transformation of the cipher can be quantified and compared, and so the model they have developed is as important of a contribution as their result of the effective extended key-length of the FX cipher, as compared to the F cipher. A final observation made by Aiello *et. al* in [1] is that the model is not only useful in measuring the effect of a cipher transform on key-search attacks, but rather it is useful in modeling the effect of any cipher-transform on any *generic attack*. A generic attack is one which does not take into account the combinatorial structure of the initial cipher, and therefore a generic attack can only exploit the structure resulting from the transform itself, and information learned from the function in a black-box model.

The Kilian-Rogaway Model

For presentation purposes we present a simplified version of the model originally presented by Kilian and Rogaway in [10].

Let $F : \{0,1\}^\kappa \times \{0,1\}^n \rightarrow \{0,1\}^n$ be a block-cipher. Let p be a polynomial, and let $TF : \{0,1\}^{p(\kappa)} \times \{0,1\}^n \rightarrow \{0,1\}^n$ be the cipher which results from applying the transformation T to F.

Definition 3.1 *A* generic-attack adversary A *is an algorithm which has* unlimited *computational power and has access to two oracles:*

- *$F(k, x)$ Oracle – The adversary supplies a $k \in \{0,1\}^\kappa$ and an $x \in \{0,1\}^n$, and the oracle returns $F(k, x)$.*

- *$E(x)$ Oracle – The adversary supplies an $x \in \{0,1\}^n$ and the oracle consistently responds* *as if it were in one of the following manners:*

World 1 – $TF_{\bar{k}}(x)$, *for a* $\bar{k} \in \{0,1\}^{p(\kappa)}$

World 2 – $\pi(x)$, *for a* $\pi \in \mathcal{P}^n$.

The adversary will be placed in one of the two worlds uniformly at random. Later we will define the security of a function transform as an upper bound on the ability of all adversaries to distinguish between which of the two worlds they were placed. However, an adversary which can query F and E at all possible locations will never have any difficulty distinguishing between worlds. Further, one might argue that in the "real world" no adversary truly has the ability to query each oracle at every point of its respective domain. In practice there are limitations on the number of chosen cipher texts that an adversary has the ability to request. This selection of chosen cipher texts corresponds to queries to the E oracle. Additionally, time restrictions would limit the effective number of queries to F which could be made.

Definition 3.2 *An* (m,t) *generic attack adversary,* $A(m,t)$*, is a generic-attack adversary,* A*, which makes exactly* m *queries of the* E *oracle, and exactly* t *queries to the* F *oracle.*

Kilian and Rogaway point out that when considering brute-force key searches we must disallow the adversary from considering any internal combinatorial structure of the primitive cipher F. This is because we are interested in the effect of the specific transformation of a cipher F on the key length, and not the weaknesses of any particular cipher to combinatorial attack. Therefore, rather than using a specific cipher we draw F uniformly at random from $\mathcal{B}$, the set of all function generators. Specifically, if we are interested in a function generator $F : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ chosen uniformly at random from set $\mathcal{B}^{\kappa,n}$ of all generators of the same form, then we can choose a random function generator F in the following manner. For each $k \in \{0,1\}^n$ we associate $f_k \in \mathcal{F}^n$, and we define $F = \{(k, f_k) | k \in \{0,1\}^\kappa\}$. Observe that by choosing F in this fashion we prevent an adversary from making use of the combinatorial structure of F by ensuring that it has no structure.

We can now formally define what it means to break the cipher TF, which is the result of applying the transform T on a cipher F.

Definition 3.3 *A generic attack adversary $A(m,t)$ is said to ϵ-*break *the TF function transform on parameters κ, n if:*

$$\left| \Pr_{F\in\mathcal{B}^{\kappa,n}, k\in\{0,1\}^{p(\kappa)}} \left[A^{TF_k, F} = 1\right] - \Pr_{F\in\mathcal{B}^{\kappa,n}, \pi\in\mathcal{P}^n} \left[A^{\pi, F} = 1\right] \right| \geq \epsilon.$$

We refer to the above difference as the advantage *of A, written ADV_A.*

Observe that this notion of breaking TF is very weak. There is no requirement of determining the key, k, nor a requirement of producing a previously unknown pair $(x, TF_k(x))$, and therefore any upper bound results in this model are very strong.

The Security of DESX

Using the model defined above Kilian and Rogaway, it is possible to show that the effective key length of a cipher $\mathrm{FX} : \{0,1\}^{2n+\kappa} \times \{0,1\}^n \rightarrow \{0,1\}^n$ resulting from the application of the DESX transform to a cipher $F : \{0,1\}^\kappa \times \{0,1\}^n \rightarrow \{0,1\}^n$ is longer than κ.

We first note that, for all t, there exists an adversary $A(m,t)$ which which can ϵ break the cipher F, for $\epsilon = \frac{t}{2^k}$ (note that breaking F corresponds to breaking TF, where T is the identity transformation). In order to $\frac{t}{2^k}$ break F, the adversary simply performs a brute force key-search. Kilian and Rogaway show the same attack cannot be as effective for the DESX transform, by proving an upper bound on the advantage of any $A(m,t)$ adversary against FX.

Theorem 3.1 (Kilian-Rogaway) *Let $A(m,t)$ be a generic attack adversary which ϵ-breaks the DESX transform applied to a cipher F with parameters κ and n, then* $\epsilon \leq \frac{t}{2^{\kappa+n-1-\log m}}$.

The theorem implies that, with the exception of adversaries which have the ability to query the E oracle on close to its entire domain, adversaries perform significantly worse on the FX then they do on F.

The Security of Composition and Triple-DES

Aiello *et. al* were able to show in [1] that under the Kilian-Rogaway model, composition increases security against generic attacks.

They define the *m-fold cascade* transform which given the cipher $F : \{0,1\}^\kappa \times \{0,1\}^n \rightarrow \{0,1\}^n$, produces the cipher $mF : \{0,1\}^{m\kappa} \times \{0,1\}^n \rightarrow \{0,1\}^n$, where $mF_{k_1 \bullet \ldots \bullet k_m}(x) = F_{k_1} \circ \cdots \circ F_{k_m}(x)$ and $|k_i| = \kappa$.

Theorem 3.2 (Aiello *et. al*) *Let $A(m,t)$ be a generic attack adversary which ϵ-breaks the m-fold cascade transform applied to a cipher F with parameters κ and n, then* $\epsilon \leq \left(\frac{t}{2^\kappa}\right)^m$.

The result is tight for $m = 2$, as a variant on the *meet in the middle* attack achieves the bound. For the the cases where $m \geq 3$, Aiello *et. al* were not able to show that the bound is tight, and they conjecture that the actual bound is much smaller.

For historical reasons Aiello *et. al* also give an upper bound for the advantage of adversaries on the triple-DES transform, which given F as a cipher outputs a cipher 3DES-F, where 3DES-$\mathrm{F}_{k_1 \bullet k_2}(x) = F_{k_1} \circ F_{k_2}^{-1} \circ F_{k_1}(x)$ and $|k_i| = \kappa$. Note that in the transform the inverse of the second function is done for historical reasons, and has no security related purpose.

Theorem 3.3 (Aiello *et. al*) *Let $A(m,t)$ be a generic attack adversary which ϵ-breaks* *the triple-DES transform applied to the cipher F with parameters κ and n, then* $\epsilon \leq \left(\frac{t}{2^\kappa}\right)^2$.

This is the same bound as for the *2-fold cascade* transform and effectively follows from the same proof, but the authors were not able to prove that this bound was tight, and they conjecture that smaller upper bounds do exist.

Chapter 4

Proposed Operators Relating to Composition and XOR

Our primary goal is to consider constructions which combine function and permutation generators into a resulting function or permutation generator which has increased security over its component generators. In particular we will be interested in constructions which use partially secure PRPGs and PRFGs to produce new PRPGs and PRFGs. Further, we want to consider constructions where the security of the resulting generator is stronger than that of its component generators, assuming none of the component generators were perfectly secure to begin with. Finally, we want the constructions to be scalable, so that as we scale the construction to be larger, we improve the security of the resulting generators.

In this chapter we will present a list of possible constructions which might be used to combine function and permutation generators in a security increasing fashion. We will consider some of the more obvious constructions which are based on the composition and XOR operators. Again, we consider these operators, as historically they have been two of the most widely used operators in the development of cryptographic systems. We will show that some constructions can be discarded as they either do not increase the security of the resulting generator, or in some cases the security of the component generators is not even maintained. In subsequent chapters we will show that three of the proposed constructions do increase the security of the resulting generator, but unfortunately we are only able to prove that one of them can be used to achieve 1-security by scaling the size of the generator.

Before describing the constructions, we will first formally describe how to combine two function generators using a generic operator, and what it means for the security to be necessarily increasing.

Definition 4.1 *Let* $\mathsf{G} = \{\mathsf{G}^n : \{0,1\}^{\ell(n)} \times \{0,1\}^n \to \{0,1\}^n | n \in \mathbb{N}\}$ *be a* $1 - \delta(n)$ *secure PRFG. Let* $\mathsf{H} = \{\mathsf{H}^n : \{0,1\}^{\kappa(n)} \times \{0,1\}^n \to \{0,1\}^n | n \in \mathbb{N}\}$ *be a* $1 - \epsilon(n)$ *secure PRFG. Let* $op = \{op^n : \{0,1\}^{r(n)} \times \mathcal{F}^n \times \mathcal{F}^n \to \mathcal{F}^n | n \in \mathbb{N}\}$ *be an operator generator, and write* $op^n(s, a, b)$ *as "*$a\ op^n_s\ b$*". Then let* $\mathsf{F} = \{\mathsf{F}^n : \{0,1\}^{\ell(n)+\kappa(n)+r(n)} \times \{0,1\}^n \to \{0,1\}^n | n \in \mathbb{N}\}$ *be the function generator defined by* $\mathsf{F}^n(k_1 \bullet k_2 \bullet k_3, x) = \left(g^n_{k_1}\ op^n_{k_3}\ h^n_{k_2}\right)(x)$*, where* $|k_1| = \ell(n)$*,* $|k_2| = \kappa(n)$ *and* $|k_3| = r(n)$*. This is written in short-hand as* $F = \mathsf{G}\ op\ \mathsf{H}$*. We say that op is security increasing if, for all polynomial time computable function generators* G *and* H*, the generator* $\mathsf{G}\ op\ \mathsf{H}$ *is a* $1 - \theta(n)$ *secure PRFG; where for some constant* $c > 0$*, and for all sufficiently large n:* $\min\{\epsilon(n), \delta(n)\} - \theta(n) > \frac{1}{n^c}$*. Similarly, we say that op is security preserving if for all sufficiently large n we know that* $\min\{\epsilon(n), \delta(n)\} - \theta(n) \leq \frac{1}{n^c}$*, for all* $c > 0$*. Finally, if op is neither security increasing or preserving then we say that it is possibly security diminishing.*

We will now present an operator and an operator generator which are are used in the constructions below.

We define the $\Box$ operator (read as box) on two functions f_1 and f_2, where f_i : $\{0,1\}^n \to \{0,1\}^n$, for $i \in \{1,2\}$, as:

$$(f_1 \Box f_2)(x) = f_1 (f_1(x) \oplus f_2(x)) \oplus f_2 (f_1(x) \oplus f_2(x)).$$

The $\Box$ operator was originally defined in some working notes by Luby and Rackoff [13].

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

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

4.1 Constructions with Known Properties

In this section we present the constructions which we were able to classify as either security increasing, preserving or possibly diminishing. We present each construction followed by an explanation of the security properties it has. In some cases the properties are stated, but the explanations are put off for a later, more in depth discussion.

In the constructions below we will assume that $\mathsf{G} = \mathsf{G}_i = \mathsf{G}_j$ for $i \neq j$. We assume that G a partially secure PRPG or PRFG, depending on the context. The indices are only there to help in describing to constructions in a clear and precise manner.

4.1.1 $F(\alpha) = \left(\mathsf{G}_{m(n)} \circ \cdots \circ \mathsf{G}_1\right)(\alpha)$

The composition of permutation generators results in a permutation generator. Therefore, we will consider the cases when the generators G_i are PRPGs and PRFGs separately. We will see that the outcomes are very different.

G is a PRPG

It was shown by Luby and Rackoff in [12] that the composition of partially secure PRPGs results in a PRPG of increased security when $m(n) = c$ for some constant c. In Chapter 5 we will discuss this result and show how to extend the result to permit any $m \in \mathcal{O}(\log\log n)$.

G is a PRFG

The composition operator is possibly security diminishing when it acts on PRFGs. The composition of a constant number of $1-\delta$ secure PRFGs, for some constant δ, can be less secure than any of its component generators. Let $\mathsf{G} = \{\mathsf{G}^n | n \in \mathbb{N}\}$ be a $1-\delta$ secure PRFG[1], where for each n the probability $\Pr_{g_k^n \in \mathsf{G}^n}[\forall x \;\; g_k^n(x) = \bar{0}] = \delta$.

For each polynomial in n sized circuit family $\{C_n\}$, for all $c > 0$ and for all sufficiently large n let

$$\left| \Pr_{g_k^n \in \mathsf{G}^n}[C_n(g^n) = 1 | \exists x \;\; g^n(x) \neq \bar{0}] - \Pr_{f^n \in \mathcal{F}^n}[C_n(f^n) = 1] \right| \leq \frac{1}{n^c}.$$

Clearly, for any function $f^n_{k_1 \bullet \cdots \bullet k_{m(n)}} = g_{k_{m(n)}} \circ \cdots \circ g_{k_1}$ if there exists an i s.t. g_{k_i} is the zero function, then h is a constant function. However, for each i, g_{k_i} is the zero function with probability δ, and indistinguishable from random with probability $1-\delta$. Therefore, if we randomly choose functions from G^n we expect to choose the first zero function on draw $\frac{1}{\delta}$. Thus, by using a simple Markov Bound we can show that for as few as $c \geq \lceil \frac{1}{\delta^2} \rceil$ composition of $1-\delta$ secure PRFGs, the construction is less than $1-\delta$ secure. Therefore, the construction is possibly security diminishing.

4.1.2 $F(\alpha) = (\mathsf{G}_{m(n)} \oplus \cdots \oplus \mathsf{G}_1)(\alpha)$

The XOR of two permutations does not result in a permutation, and therefore we will only consider the construction when G is a partially secure $PRFGs$. We show that the XOR operator is at best security preserving.

Let $\widehat{\mathsf{G}} = \{\widehat{\mathsf{G}}^n : \{0,1\}^{k(n)} \times \{0,1\}^n \to \{0,1\}^n\}$ be a PRFG. We modify $\widehat{\mathsf{G}}$ to form G as follows: for each n and each $\kappa \in \{0,1\}^{k(n)}$ we set $g_\kappa^n = \hat{g}_\kappa^n$ and then we set the first bit of $g_\kappa^n(\bar{0})$ to be 0. Notice that G is $\frac{1}{2}$-secure, as the $\Pr_{g_\kappa^n \in \mathsf{G}^n}(g_\kappa^n(\bar{0})'s$ first bit is $0) = 1$, while

[1]A PRFG such as G is constructed by taking a 1-secure PRFG and modifying it so that the first δ fraction of the keys correspond to the zero function, and the remaining keys correspond to the same function as they did in the 1-secure PRFG.

$\Pr_{f^n \in \mathcal{F}^n}(f^n(\bar{0})'s$ first bit is $0) = \frac{1}{2}$. However, for all $\tilde{g} \in \mathsf{G} \oplus \mathsf{G}$ it is the case $\tilde{g}(\bar{0})$'s first bit is 0. Therefore, it is clear that if we let $\mathsf{G}_i = \mathsf{G}$, for all i, then for all c the construction $\mathsf{G}_c \oplus \cdots \oplus \mathsf{G}_1$ is still $\frac{1}{2}$ secure.

Unfortunately this example is not very robust. In particular, the example clearly generalizes for PRFGs which are δ secure, where $\delta = 2^{-i}$ for some positive integer i. However, we are not aware of any examples of PRFGs which are $1-\delta$ secure when $\delta = 2^{-i}$, which when used in the construction above are security preserving. All examples we have tried have been security increasing. This leaves us with the following question.

Open Question 4.1 *Is the construction $F(\alpha) = (\mathsf{G}_{m(n)} \oplus \cdots \oplus \mathsf{G}_1)(\alpha)$ security increasing, when the generators G_i are restricted to being $1-\delta$ secure for $\delta < \frac{1}{2}$.*

4.1.3 $F(\alpha) = (\mathsf{G}_{m(n)} \square (\mathsf{G}_{m(n)-1} \square (\cdots \mathsf{G}_3 \square (\mathsf{G}_2 \square \mathsf{G}_1) \cdots)))(\alpha)$

When the $\square$ operator is applied to two permutations the resulting generator is not a permutation, and so we consider only the case in which G is a PRFG. The $\square$ operator is a security increasing operator, when applied to PRFGs. The proof of this is shown in Chapter 6, and so we do not discuss it further here.

4.1.4 $F(\alpha) = (\mathsf{G}_{2m(n)} \square \mathsf{G}_{2m(n)-1}) \circ \cdots \circ (\mathsf{G}_2 \square \mathsf{G}_1)(\alpha)$

For the same reason as in section 4.1.3, we consider only the case in which G is a PRFG. This construction is not even security preserving for PRFGs, for the same reasons as those presented in section 4.1.1.

Let $\mathsf{G} = \{\mathsf{G}^n | n \in \mathbb{N}\}$ be a $1-\delta$ secure PRFG, where for each n the probability $\Pr_{g_k^n \in \mathsf{G}^n}[\forall x \;\; g_k^n(x) = \bar{0}] = \delta$. For each polynomial in n sized circuit family $\{C_n\}$, for all $c > 0$, and for all sufficiently large n let

$$\left| \Pr_{g_k^n \in \mathsf{G}^n}[C_n(g^n) = 1 | \exists x \;\; g^n(x) \neq \bar{0}] - \Pr_{f^n \in \mathcal{F}^n}[C_n(f^n) = 1] \right| \leq \frac{1}{n^c}.$$

Notice that with probability δ^2 the generator $(\mathsf{G}_{i+1}\Box\mathsf{G}_i)$ is the zero function. Now following the same argument made in section 4.1.1, we see that the security of the construction for size $m(n) \geq \lceil\frac{1}{\delta^4}\rceil$ is less than $1-\delta$ secure. Therefore, the construction is possibly security diminishing.

4.1.5 $\mathsf{F}(\alpha) = \mathsf{G}_{m(n)}\Diamond\cdots\Diamond\mathsf{G}_1(\alpha)$

When the $\Diamond$ operator generator is applied to two permutations the result is not a permutation, and so we consider only the case in which G is a PRFG. The $\Diamond$ operator generator is a security increasing operator generator, when applied to PRFGs. The proof of this is shown in Chapter 7, and so we do not discuss it further here.

4.2 Constructions with Unknown Properties

Below we give a list of constructions for which we are unaware of their status as either security increasing, preserving or possibly diminishing. However, for all but one of the constructions, we will give reasons to suggest that these operators are not interesting, and therefore not worth future study. We list the constructions below, and then discuss them.

1. $\mathsf{F}(\alpha) = \left((\mathsf{G}_{2m(n)} \circ \mathsf{G}_{2m(n)-1}) \oplus \cdots \oplus (\mathsf{G}_2 \circ \mathsf{G}_1)\right)(\alpha)$
2. $\mathsf{F}(\alpha) = \left((\mathsf{G}_{2m(n)}\Box\mathsf{G}_{2m(n)-1}) \oplus \cdots \oplus (\mathsf{G}_2\Box\mathsf{G}_1)\right)(\alpha)$
3. $\mathsf{F}(\alpha) = \left((\mathsf{G}_{2m(n)} \oplus \mathsf{G}_{2m(n)-1}) \circ (\mathsf{G}_{2m(n)-2} \oplus \mathsf{G}_{2m(n)-3}) \circ \cdots \circ (\mathsf{G}_4 \oplus \mathsf{G}_3) \circ (\mathsf{G}_2 \oplus \mathsf{G}_1)\right)(\alpha)$

We suspect that construction 1 in the list above is probably security increasing, for it is similar in concept to the construction in section 4.1.5 which uses the $\Diamond$ operator. This is because we can view the expression $\mathsf{H}\Diamond\mathsf{G}$ as $(\mathsf{H} \circ \mathsf{R}) \oplus (\mathsf{G} \circ \mathsf{R})$ where $\mathsf{R}^n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ is a generator, where for all $k \in \{0,1\}^n$ we define $r_k \in \mathsf{R}^n$ as $r_k(x) =$

$x \oplus k$. However, the effect of construction 1 on security is probably weaker than that of the construction presented in section 4.1.5. Further, the constructions would have slower running times, and be harder to implement. Finally, due to the structure of the constructions, we believe that any proof that the construction is security increasing would be much more technically complicated then the proof which will be presented for construction 4.1.5. Therefore, we suspect that further research into these construction will fail to provide any insight which has not previously been seen in considering other constructions, and thus we will not mention them further in this work.

Progress in determining the security related properties of construction 2 from the list has been stifled for many of the same reasons we were stifled from making progress in answering Open Question 4.1. We suggest that the solution to this problem is closely related to the solution of previously mentioned open question.

Open Question 4.2 *Is the construction* $\mathsf{F}(\alpha) = \left((\mathsf{G}_{2m(n)} \square \mathsf{G}_{2m(n)-1}) \oplus \cdots \oplus (\mathsf{G}_2 \square \mathsf{G}_1) \right)(\alpha)$ *security increasing, security preserving or security decreasing.*

Finally, construction 3 may be slightly security increasing. However, we will show that, even if the construction is security increasing, there is maximum security of $1-\delta^2$ which can be achieved if G is a $1-\delta$ secure generator. Therefore, we cannot use this construction to try and create 1 secure generators, and thus it is of limited use.

To observe the upper bound of $1-\delta^2$ security, we define $\mathsf{G} = \{\mathsf{G}^n | n \in \mathbb{N}\}$ to be a $1-\delta$ secure PRFG, where for each n the probability $\Pr_{g_k^n \in \mathsf{G}^n}[\forall x \;\; g_k^n(x) = \bar{0}] = \delta$. For each polynomial in n sized circuit family $\{C_n\}$, for all $c > 0$, and for all sufficiently large n let

$$\left| \Pr_{g_k^n \in \mathsf{G}^n}[C_n(g^n) = 1 | \exists x \;\; g^n(x) \neq \bar{0}] - \Pr_{f^n \in \mathcal{F}^n}[C_n(f^n) = 1] \right| \leq \frac{1}{n^c}.$$

We observe that with probability δ^2 the generator $\mathsf{G} \oplus \mathsf{G}$ is the zero function. Therefore with probability δ^2 the generator $(\mathsf{G}_{2m(n)} \oplus \mathsf{G}_{2m(n)-1})$ is the zero function, and therefore with probability at least δ^2 construction 3 is the zero function. Therefore, the security of the construction can never be higher than $1 - \delta^2$, and thus we cannot use it to develop a 1-secure generator.

Chapter 5

The Composition Operator

Luby and Rackoff [12] were the first to prove that the composition of two partially secure PRPGs results in a PRPG which has stronger security than either of the initial primitives. In [12] they gave an outline of the proof, and later Akcoglu and Rackoff in [2] filled in the missing details. It would be convenient if, by inductively repeating the argument many times, it could shown that a partially secure PRPG could be composed with itself many times, and result in a completely secure PRPG. Unfortunately due to technical restrictions, which will be explained in the sequel, the given proof only works for a constant number of compositions.

In this chapter we will demonstrate that it is possible to modify the argument presented by Luby and Rackoff to permit a non-constant number of compositions. We will first give the framework for the original result by Luby and Rackoff and explain its limitation to a constant number of compositions. Next, we will extend this result to allow a $\mathcal{O}(\log \log n)$ number of compositions, and in doing so we basically outline the proof given by Luby and Rackoff, but make it less dependent on sampling. This will result in a more secure construction then was possible under the Luby-Rackoff result.

5.1 The Luby-Rackoff Result

The main result of Luby and Rackoff is to show that the composition of two partially secure PRPGs results in a PRPG which is provably more secure then either of the component generators. We present their theorem in a slightly different manner than it was originally presented in [12] or in the unpublished manuscript [2]. This presentation makes the limitations of their results more immediate, and is comparable to the presentation of Levin's proof of Yao's XOR lemma in [6].

Lemma 5.1 (Composition Isolation Lemma – Luby & Rackoff's Version) – *There exist fixed polynomials p_1 and p_2 such that for all $0 \le \epsilon, \delta \le 1$; polynomials $c_{\mathsf{G}}, c_{\mathsf{H}}$, and s_{F}; and permutation generators H and G, where $c_{\mathsf{G}}(n)$ and $c_{\mathsf{H}}(n)$ bound from above the size of the circuits which compute G^n and H^n respectively. Define $\mathsf{F} = \mathsf{G} \circ \mathsf{H}$.*
Hypothesis: *If there exists a family of decision-circuits $\{C_n\}$, where for each n the circuit C_n is of size $s_{\mathsf{F}}(n)$, and for some $c > 0$ and infinitely many n:*

$$\left|\Pr_{C_n}(\mathsf{F}^n) - \Pr_{C_n}(\mathcal{F}^n)\right| \ge \epsilon\delta\,(2 - \max\{\epsilon, \delta\}) + \frac{1}{n^c}.$$

Conclusion: *Then for infinitely many n there exists either a decision-circuit Λ_n of size $p_1(n^c \cdot c_{\mathsf{G}}(n))s_{\mathsf{F}}(n)$ for which:*

$$\left|\Pr_{\Lambda_n}(\mathsf{G}^n) - \Pr_{\Lambda_n}(\mathcal{F}^n)\right| \ge \delta + \frac{1}{n^{24c}};$$

or a decision-circuit Υ_n of size $p_2(n^c \cdot c_{\mathsf{G}}(n))s_{\mathsf{F}}(n)$ for which:

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

or a decision-circuit Ξ_n of size $c_{\mathsf{H}}(n) + s_{\mathsf{F}}(n)$ for which:

$$\left|\Pr_{\Xi_n}(\mathsf{G}^n) - \Pr_{\Xi_n}(\mathcal{F}^n)\right| \ge \delta + \frac{1}{n^{6c}}.$$

We can now show that by composing a partially secure PRPG with itself a constant number of times we get a significantly more secure PRPG. This result follows from repetitively applying the Isolation Composition Lemma. We state this formally in the corollary below.

Theorem 5.1 (Composition Theorem – Luby & Rackoff's Version) *Let* G *be a* $1-\epsilon$ *secure PRPG. Then for each positive integer* c*, the generator* $\mathsf{F} = \underbrace{\mathsf{G} \circ \cdots \circ \mathsf{G}}_{c \text{ times}}$ *is* $1-\theta$ *secure, where* $\theta = \epsilon^c(2-\epsilon)^{c-1}$.

Proof(Sketch): Let $\{C_n\}$ be a family of circuits. For each n let the size of C_n be less than $s_{\mathsf{F}}(n)$, for a polynomial s_{F}. Let C_{G} and C_{H} be polynomials which bound the size of the circuits needed to compute G^n and H^n respectively. Assume that, for infinitely many n, C_n can distinguish between a random function in $\mathcal{F}^n$ and a pseudo-random permutation in F^n with probability at least $\epsilon^c(2-\epsilon)^{c-1} + \frac{1}{n^d}$, for some constant d.

Since $\mathsf{F} = \underbrace{\mathsf{G} \circ (\mathsf{G} \circ \cdots \circ \mathsf{G})}_{c \text{ times}}$, we know by the Composition–Isolation Theorem that there exits polynomials p_2 and p_1; and either a circuit of size $\max\left\{s_{\mathsf{F}}(n)p_1\left(c_{\mathsf{G}}(n)n^d\right), s_{\mathsf{F}}(n) - c_{\mathsf{H}}(n)\right\}$ which ϵ distinguishes G; or a circuit of size $\left(s_{\mathsf{F}}(n)p_2(c_{\mathsf{G}}(n)n^d)\right)$ which $\epsilon^{c-1}(2-\epsilon)^{c-2}$ distinguishes $\underbrace{(\mathsf{G} \circ \cdots \circ \mathsf{G})}_{c-1 \text{ times}}$. Since the former would contradict the fact that G is $(1-\epsilon)$ secure, it must be the case that there is a circuit which $\epsilon^{c-1}(2-\epsilon)^{c-2}$ distinguishes $\underbrace{(\mathsf{G} \circ \cdots \circ \mathsf{G})}_{c-1 \text{ times}}$. We can now re-apply this argument inductively $c-2$ more times to show that there exists a polynomial p_2' and a circuit of size no larger than $s_{\mathsf{F}}(n)\left(p_2'(c_{\mathsf{G}}(n)n^{d'})\right)^{c-1}$, for some constant d', which ϵ distinguishes G from random functions. This contradicts the claim that G is $1-\epsilon$ secure, and completes the proof. ...$\Box$

Notice that the proof fails when the number of compositions, c, becomes non-constant. This is because the circuit which ϵ distinguishes G_c is of size $s_{\mathsf{F}}(n)\left(p_2'(c_{\mathsf{G}}(n)n^{d'})\right)^{c-1}$, but when c is constant this functions is asymptotically larger than any polynomial. It is this limitation which prevents us from attaining a PRPG which is completely secure using the Isolation Lemma. In the sequel we will show that this limitation can be partially overcome, but while more security is attained the goal of complete security is still elusive.

A natural question to contemplate is whether the upper bound of distinguishing F from $\mathcal{F}^n$ is tight in the Isolation Lemma. Perhaps there are smaller bounds which could replace the bound of $\epsilon\delta(2-\max\{\epsilon,\delta\})$, and otherwise leave the lemma unchanged. In the next section we show that the Isolation Lemma is tight with respect to this bound.

5.1.1 The Isolation Lemma is Tight

We show that there exist PRPGs H and G which are respectively $(1-\epsilon)$ and $(1-\delta)$ secure[1], but when composed together the result is exactly $(1-\theta)$ secure, where $\theta = \epsilon\delta(2-\max\{\epsilon,\delta\})$.

The Construction of H and G

To simplify the presentation we assume that δ and ϵ are of the form $\frac{1}{2^c}$ or $1-\frac{1}{2^c}$, for some constant c. Let $\widehat{\mathsf{G}} = \{\widehat{\mathsf{G}}^n : \{0,1\}^{\ell(n)} \times \{0,1\}^n \to \{0,1\}^n | n \in \mathbb{N}\}$ be a PRPG. We construct $\mathsf{G} = \{\mathsf{G}^n : \{0,1\}^{\ell(n)+c} \times \{0,1\}^n \to \{0,1\}^n | n \in \mathbb{N}\}$ to be $1-\delta$ secure in two steps. We will describe the construction of G^n, and note that the construction of H^n is similar.

First we set a fraction δ of the keys to correspond to the identity permutation, and the remainder to correspond to permutations chosen from $\widehat{\mathsf{G}}^n$. This is done in two different fashions dependent on the form of δ as described below:

Case 1 ($\delta = 1-\frac{1}{2^c}$) For each $k \in \{0,1\}^{\ell(n)+c}$ we set the permutation $g_k^n \in \mathsf{G}^n$ to be the identity permutation if the first c bits of k *are not* all 0; otherwise let $\bar{k}$ be the last $\ell(n)$ bits of k, and set $g_k^n = \hat{g}_{\bar{k}}^n$, for $\hat{g}_{\bar{k}}^n \in \widehat{\mathsf{G}}^n$.

Case 2 ($\delta = \frac{1}{2^c}$) For each $k \in \{0,1\}^{\ell(n)+c}$ we set the permutation $g_k^n \in \mathsf{G}^n$ to be the

[1]Of course we can't actually show that such generators exits, but we show that if PRPGs exist, then H and G exist.

identity permutation if the first c bits of k *are* all 0; otherwise let $\bar{k}$ be the last $\ell(n)$ bits of k, and set $g_k^n = \hat{g}_{\bar{k}}^n$, for $\hat{g}_{\bar{k}}^n \in \widehat{\mathsf{G}}^n$.

The second step in the construction of G^n is to ensure that for each $k \in \{0,1\}^{\ell(n)+c}$ we set the value of $g_k^n(\bar{0})$ in one of two manners, depending on the form of δ. We describe this transformation below:

Case 1 ($\delta = 1 - \frac{1}{2^c}$) For all k set the first c bits of $g_k^n(\bar{0})$ to 0; the last $n - c$ bits remain as they were in $g_k^n(\bar{0})$.

Case 2 ($\delta = \frac{1}{2^c}$) For all k ensure that not all of the first c bits of $g_k^n(\bar{0})$ are 1. If they are, set them to be a member of the set $\{0,1\}^c \setminus \{1^c\}$ chosen uniformly at random[2]; the last $n - c$ bits remain as they were in $g_k^n(\bar{0})$.

Notice that in order to maintain the permutation property of G^n we can simply store both the initial and modified value of $g_k^n(\bar{0})$. Should a query ever be made to the initial preimage of the modified value of $g_k^n(\bar{0})$, then we respond with the initial value of $g_k^n(\bar{0})$.

It remains to show that G and H are in fact $1 - \delta$ and $1 - \epsilon$ secure, and we refer the interested reader to Appendix A for a proof of this claim.

The Adversary

We now show that there exists an adversary which can $\epsilon\delta(2 - \max\{\epsilon, \delta\})$ distinguish $\mathsf{F} = \mathsf{G} \circ \mathsf{H}$, and thereby show that the Isolation Lemma is tight. We assume that $\delta \geq \epsilon$. A similar argument to the one presented handles the case when $\epsilon > \delta$.

Given a function $w \in \mathcal{F}^n$ the adversary, A, accepts in one of the following two conditions, dependent on the form of δ:

[2]In practice a PRNG p would be used to set the first c bits of $g_k^n(\bar{0})$. This would be done by setting the first c bits of $g_k^n(\bar{0})$ to be equivalent be the first consecutive set of c bits in $p(k)$ which were not all 1. This allows the first c bits to be computed, and discards the need to store them, which would not be possible as it would require an exponential amount of storage.

Case 1 ($\delta = 1 - \frac{1}{2^c}$) The adversary A accepts *iff* the first c bits of $w(\bar{0})$ are 0;

Case 2 ($\delta = \frac{1}{2^c}$) The adversary A accepts *iff* the first c bits of $w(\bar{0})$ are not 1.

We first consider the accepting probability of A if w was chosen uniformly from F^n. In this case w is the composition of two permutations chosen randomly from each of the respective generators G^n and H^n. We partition the permutations of G^n into two sets. Let I_{G^n} represent the set of all identity permutations in G^n, and let P_{G^n} represent the remaining permutations in G^n. Similarly, partition H^n into two sets I_{H^n} and P_{H^n}. We now consider the probability that w is accepted based on it being the composition of two permutations from the above mentioned partitions.

Case 1 ($w \in I_{\mathsf{G}^n} \circ I_{\mathsf{H}^n}$) The adversary will accept w with probability 1, and the case occurs with probability $\delta\epsilon$.

Case 2 ($w \in I_{\mathsf{G}^n} \circ P_{\mathsf{H}^n}$) The adversary will accept w with probability $\frac{1-\delta}{1-\epsilon}$. This is the probability that A would accept a random function conditioned on the fact that w was chosen from P_{H^n}. The case occurs with probability $\delta(1-\epsilon)$.

Case 3 ($w \in P_{\mathsf{G}^n} \circ I_{\mathsf{H}^n}$) The adversary will accept w with probability 1, and the case occurs with probability $(1-\delta)\epsilon$.

Case 4 ($w \in P_{\mathsf{G}^n} \circ P_{\mathsf{H}^n}$) The adversary will accept w with probability $1-\delta$. This is the probability that the result of a random function evaluated at a random point in its range would meet the acceptance criteria of A. The case occurs with probability $(1-\delta)(1-\epsilon)$.

Clearly a random permutation w will be accepted by A with probability $1-\delta$. Allowing X to be $\max\{\epsilon, \delta\}$, and Y to be $\min\{\epsilon, \delta\}$, we see that $X = \delta$ and $Y = \epsilon$. We now see that the distinguishing probability of A on F is:

$$\epsilon\delta + \frac{(1-\delta)}{(1-\epsilon)}\delta(1-\epsilon) \quad + \quad (1-\delta)\epsilon + (1-\delta)^2(1-\epsilon) - (1-\delta)$$

$$
\begin{aligned}
&= \big[YX + (1-X)X + (1-X)Y + (1-X)^2(1-Y)\big] - (1-X) \\
&= [(1-X) + XY(2-X)] - (1-X) \\
&= \epsilon\delta\,(2 - \max\{\epsilon,\delta\})\,.
\end{aligned}
$$

5.2 The Improved Composition Lemma

As we've just seen, we cannot improve the composition lemma by improving the security parameter of the resulting PRPG, in the Isolation Lemma. But, in this section we will prove that if we permit a slightly more liberal notion of security for our partially secure PRPGs, then we can prove a lemma similar to the Isolation Lemma. Further, this lemma permits more than a constant number of compositions of PRPGs. The result is that a non-constant number of compositions of a partially secure PRPG, under the liberal notion of security, gives a PRPG which is provably more secure then the previously stated results.

5.2.1 Semi-Secure PRPGs

For the remainder of this chapter we will call a PRPG, H, $1-\epsilon$ *semi-secure*, if there exists no family of polynomial-sized circuits $\{C_n\}$ which can $\epsilon + \frac{1}{\log^c n}$ distinguish H from random, for some constant c and infinitely many n.

Note that a generator H which is $1-\epsilon$ secure is also $1-\epsilon$ semi-secure. However, the converse need not be true. The reason for the relaxation of the definition of security is due to the observation that the blow-up in circuit size which occurs in the Luby-Rackoff Isolation Lemma is due to a large amount of sampling which must be done on various distributions which occur in the proof. However, the amount of sampling that must be done is inversely proportional to the leniency we permit the adversary in ϵ-distinguishing a PRPG from the random set of functions in order for the generator to still be called $1-\epsilon$ secure. Therefore, by making the security definition slightly weaker we reduced the amount of sampling required for the proof to hold; which reduces the size of the circuits constructed in the proof of the isolation lemma; which permits a greater number of compositions in the composition lemma; which results in a PRPG with stronger security then was possible under the original isolation lemma.

Finally observe that if, for any constant $0 < \epsilon < 1$, we are given a $1 - \epsilon$ semi-secure generator F, then the same generator is definitely $1 - \sqrt{\epsilon}$ secure. However, by composing the generator with itself we get a $1 - (2 \cdot \epsilon - \sqrt{\epsilon})$ secure generator, which is strictly more secure than the a $1 - \epsilon$ secure generator. This result follows from the original Luby-Rackoff Isolation Lemma, and allows us to transform many –but definitely not all– of the security results under the semi-secure definition to corresponding results under the normal security definition.

Now that we have motivated the semi-secure definition we present the new version of the Isolation Lemma which is less dependent on sampling.

5.2.2 A Stronger Isolation Lemma

Lemma 5.2 (Composition Isolation Lemma – Stronger Version) – *There exists a fixed polynomial p_1 and a fixed poly-logarithmic function p_2 such that for all $0 < \delta < 1$; $\epsilon : \mathbb{N} \to [0,1]$, where for all sufficiently large n, $\epsilon(n) < \delta$; polynomials $c_{\mathsf{G}}, c_{\mathsf{H}}$, and s_{F}; and permutation generators H and G, where $c_{\mathsf{G}}(n)$ and $c_{\mathsf{H}}(n)$ bound from above the size of the circuits which compute G^n and H^n respectively. Define $\mathsf{F} = \mathsf{G} \circ \mathsf{H}$.*

Hypothesis: *If there exists a family of decision-circuits $\{C_n\}$, where for each n the circuit C_n is of size $s_{\mathsf{F}}(n)$, and for some $c > 0$ and infinitely many n:*

$$\left| \Pr_{C_n}(\mathsf{F}^n) - \Pr_{C_n}(\mathcal{F}^n) \right| \geq \epsilon(n)\delta\left(2 - \max\{\epsilon, \delta\}\right) + \frac{1}{\log^c n}.$$

Conclusion: *Then for infinitely many n:*

there exists either a decision-circuit Λ_n of size $p_1(\log^c n \cdot c_{\mathsf{G}}(n))s_{\mathsf{F}}(n)$ for which,

$$\left| \Pr_{\Lambda_n}(\mathsf{G}^n) - \Pr_{\Lambda_n}(\mathcal{F}^n) \right| \geq \delta + \frac{1}{n};$$

or a decision-circuit Υ_n *of size* $p_2(\log^c n \cdot c_{\mathsf{G}}(n))s_{\mathsf{F}}(n)$ *for which,*

$$\left|\Pr_{\Upsilon_n}(\mathsf{H}^n) - \Pr_{\Upsilon_n}(\mathcal{F}^n)\right| \geq \epsilon(n) + \frac{1}{\log^{2c} n};$$

or a decision-circuit Ξ_n *of size* $c_{\mathsf{H}}(n) + s_{\mathsf{F}}(n)$ *for which,*

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

The interesting difference between the two Isolation Lemmas occurs in the differing definitions of p_2. The new definition of p_2 increases the upper bound for distinguishing between H^n and random functions. This is what necessitates the definition of semi-security. Further, the new definition of p_2 significantly affects the size of the circuit which can distinguish H^n from $\mathcal{F}^n$. The differing circuit size permits a stronger version of the Composition Theorem, which we will present later.

Another difference to observe is that, in the original Isolation Lemma the results were symmetric. By this we mean that given generators $\widehat{\mathsf{H}}$ and $\widehat{\mathsf{G}}$, the Lemma applied equally to both of the constructions $\mathsf{F}_1 = \widehat{\mathsf{H}} \circ \widehat{\mathsf{G}}$ and $\mathsf{F}_2 = \widehat{\mathsf{G}} \circ \widehat{\mathsf{H}}$. This can be observed by simply renaming the generators $\widehat{\mathsf{G}}$ and $\widehat{\mathsf{H}}$. Notice that a similar result does not follow from the statement of the strong version of the Isolation Lemma. We note that it is possible to prove a symmetric version of this Lemma by using symmetric versions of the arguments presented in this chapter. Unfortunately, there is one exception, for it seems that in order to prove a symmetric version of one of the lemmas used in the proof, one must first prove the lemma as it is presented in this chapter, and then use this result to prove the lemma's symmetric version. We will discuss this further in the sequel.

Theorem 5.2 (Composition Theorem – Stronger Version) –
Let $0 < \epsilon < 1$*, and let* G *be a* $1-\epsilon$ *secure PRPG. Then for each* $f \in \mathcal{O}(\log\log n)$ *the generator* $\mathsf{F} = \underbrace{\mathsf{G} \circ \cdots \circ \mathsf{G}}_{f(n)\ times}$ *is* $1-\theta(n)$ *semi-secure, where* $\theta(n) = \epsilon^{f(n)}(2-\epsilon)^{f(n)-1}$.

Proof: Similar to the proof of Theorem 5.1. ...□

Observe that this theorem shows that given some generator which is $1-\epsilon$ secure, where $\epsilon = \frac{1}{c}$ for some $c > 0$, then by performing $\mathcal{O}(\log\log n)$ compositions of the generator we can achieve a $1 - \frac{1}{\log^d n}$ *semi-secure* generator, for any constant $d > 0$. We then observe the fact that a $1 - \frac{1}{\log^d n}$ semi-secure secure generator is $1 - \frac{1}{\log^{d-1} n}$ secure. Therefore by performing $\mathcal{O}(\log\log n)$ compositions of the generator we can achieve a $1 - \frac{1}{\log^{d'} n}$ *secure* generator, for any constant $d' > 0$.

Before presenting the proof of Theorem 5.2 we cite some lemmas used in the proof which are either commonly seen in the literature or which have previously appeared in the manuscript of Akcoglu and Rackoff [2].

Lemma 5.3 (Chernoff bound –(Weak)) *Let $x_1, .., x_{\log^t n}$ be i.i.d.r.v. which take the values 0 or 1 with probabilities q or $p = 1 - q$ respectively. Let $X_{\log^t n} = \frac{1}{\log^t n}\sum_{i=1}^{\log^t n} x_i$. Then for any k and l, there exists a t such that:*

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

Lemma 5.4 (Chernoff Bound–(Strong)) *Let $x_1, .., x_{n^t}$ be i.i.d.r.v. which take the values 0 or 1 with probabilities q or $p = 1 - p$ 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 s.t.:*

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

Both versions of the Chernoff Bound follow immediately from the standard proof of the Chernoff bound. For a proof of the Chernoff bound see [17] or [19]. ...□

As mentioned in section 2.1, there is no need to have probabilistic circuits as adversaries, as we can always derandomize them. This is done by fixing the random input bits of the circuit to be a specific string of bits which "does well" on the average input, where "does well" is defined relative to how we are attempting to use the circuit. Next we will present a lemma which is a formalization of a similar but more complicated idea.

In this case we have a probabilistic circuit which "does well" on two different distributions of inputs (note these are distributions over the inputs of the circuit, and not the distributions over the strings of random bits which are used by the circuit). We will show that that there exists a specific string which can be fixed as the random input bits of the circuit, so that the circuit is now guaranteed to do "almost as well" on both distributions of inputs, as it originally did on each distribution individually, when the circuit was still probabilistic.

Lemma 5.5 (Derandomization Lemma) *Let $C_n(w,r)$ be a probabilistic oracle-circuit, where $w \in \mathcal{F}^n$ represents the oracle function, and r represents C_n's random input bits. Let D_1^n and D_2^n be two distributions over $\mathcal{F}^n$, and let R_n be the distribution over C_n's random bits. Let $P : \mathcal{F}^n \times \mathbb{R} \to \{0,1\}$ be a predicate. Then, If*

$$\Pr_{w\in D_1^n,\, r\in R_n}[P(w, C_n(w,r)) = 1] \geq 1 - p \quad \textit{and} \quad \Pr_{w\in D_2^n,\, r\in R_n}[P(w, C_n(w,r)) = 1] \geq 1 - p,$$

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

Intuitively, in the lemma above, the circuit C is trying to compute a value for the oracle w. However, since C is probabilistic, the value of the circuit on input w may be different for different strings of random bits. Therefore, we measure whether or not the circuit has computed an acceptable value, given the oracle w, by using the predicate P, which indicates, on all pairs (w,v), whether v is an acceptable value for C to output, given the oracle w. For an example of this lemma's use see section 5.3.1.

Proof: We build a third distribution D_3^n by first choosing a random random bit $i \in \{1,2\}$, and then choosing a random function $w \in D_i$. Note that

$$\Pr_{w\in D_3^n,\, r\in R_n}[P(w, C_n(w,r)) = 1] \geq \frac{1}{2}(1-p) + \frac{1}{2}(1-p) = 1 - p,$$

and so there exists an $\tilde{r} \in R_n$, for which

$$\Pr_{w\in D_3^n}[P(w, C_n(w,\tilde{r})) = 1] \geq 1 - p. \tag{5.1}$$

Now it must be the case that $\Pr_{w\in D_i^n}[P(w, C_n(w,\tilde{r})) = 1] \geq 1-2p$, for $i \in \{1,2\}$; otherwise $D_3^n[P(w, C_n(w,\tilde{r})) = 1] < \frac{1}{2}(1-2p)+\frac{1}{2} = 1-p$, and this contradicts (5.1).

...□

Lemma 5.6 *For any $S \subset \mathcal{P}^n$ and any decision-circuit C_n the* $\Pr_{C_n}(\mathcal{F}^n \circ S) = \Pr_{C_n}(\mathcal{F}^n)$.

Proof: Follows from the fact that each element of S is a permutation, and therefore the resulting distribution on the queries to $\mathcal{F}^n$ is invariant under when it is composed with the set S. ...□

Lemma 5.7 (Luby-Rackoff) *For each n, let $S^n \subset \mathcal{F}^n$ such that $\frac{|S^n|}{|\mathcal{F}^n|} \geq \frac{1}{n^d}$, for some constant d. Then, for any constant k, for any polynomial-sized decision-circuit family $\{C_n\}$, and for all but a $\frac{1}{2^{n/3}}$ fraction of $\nu \in \mathcal{F}^n$:*

$$|\Pr_{C_n}(\nu \circ S^n) - \Pr_{C_n}(\mathcal{F}^n)| < \frac{1}{n^k},$$

for sufficiently large n.

This lemma follows directly from the lemma stated bellow, and the strong version of the Chernoff Bound.

Lemma 5.8 *Let S^n and $\{C_n\}$ be as in Lemma 5.7. Then, given k there is an r such that for all sufficiently large n:*

$$\Pr_{\nu, s_1, \ldots, s_{n^r} \in \mathcal{F}^n \times (S^n)^{n^r}} \left(\left| \frac{1}{n^r} \sum_{i=1}^{n^r} C_n(\nu \circ s_i) - \Pr_{C_n}(\mathcal{F}^n) \right| > \frac{1}{n^k} \right) < \frac{1}{2^{n/3}}.$$

This lemma is proven in [2], and a complete proof of similar lemmas will be presented in Chapters 6 and 7 (See the proofs of Lemmas 6.3 and 7.3). Therefore, we will only give a sketch of the lemma's proof.

Proof(Sketch): Observe that whenever we have a circuit C_n which has oracle-gates which correspond to a function f, it can be easily modified, by adding at most a polynomial number of gates and connections between gates, to render a new circuit $\widehat{C_n}$, where

$\widehat{C_n}$ never repeats queries to f on the same input. Further, any polynomial sized family of circuits $\{C_n\}$, in which each circuit C_n has at most $m(n)$ oracle gates, can be easily be modified to a equivalent family of circuits $\{\bar{C_n}\}$, where each circuit $\bar{C_n}$ has exactly $m(n)$ oracle gates. We shall assume that all circuits in this proof sketch are of the modified forms described above.

We perform two experiments. In the first we randomly choose $f_1, ..., f_{n^r} \in (\mathcal{F}^n)^{n^r}$, and compute the average value, E_1, of the circuit on these function. Specifically we let $E_1 = \frac{1}{n^r}\sum_{i=1}^{n^r} C_n(f_i)$. In the second experiment we randomly choose $\nu, s_1, ..., s_{n^r} \in \mathcal{F}^n \times (S^n)^{n^r}$, and compute the average value, E_2, of the circuit on the functions $\nu \circ s_i$. Specifically we let $E_2 = \frac{1}{n^r}\sum_{i=1}^{n^r} C_n(\nu \circ s_i)$. Using the strong version of the Chernoff Bound we see that the probability that $|E_1 - \Pr_{C_n}(\mathcal{F}^n)| > \frac{1}{n^k}$ is less than $2^{n/2}$. We then show that the difference between E_1 and E_2 is bounded to be less than $2^{n/2}$, and the result follows.

To bound the difference between E_1 and E_2 we notice that from the perspective of the circuit C_n that the functions $(f \circ s_1), ..., (f \circ s_{n^r})$ will appear to be random functions chosen uniformly from $\mathcal{F}^n$, *so long as f is never queried on the same input throughout experiment two.* Since each circuit never performs the same oracle query twice, we know that for each i, the function f will not be queried twice on the same input during the evaluation of $C_n(f \circ s_i)$. But for $i \neq j$, it may be the case that during the evaluations of $C_n(f \circ s_i)$ and $C_n(f \circ s_j)$, that the function f is queried on the same input twice. However, we can bound this probability. WLOG, assume that $i < j$, and let α be a query made during the evaluation of $C_n(f \circ s_i)$. Let β_k to be the k^{th} query which is made during the evaluation of $C_n(f \circ s_j)$. If we fix s_i and α, then the probability that $s_i(\alpha) = s_j(\beta_k)$ is:

$$\Pr_{s_j \in S^n}[s_i(\alpha) = s_j(\beta_k)] = \Pr_{s_j \in \mathcal{F}^n}[s_i(\alpha) = s_j(\beta_k)|s_j \in S^n] \leq \frac{\frac{1}{2^n}}{\frac{1}{n^d}} = \frac{n^d}{2^n},$$

where the $\frac{1}{n^d}$ in the last inequality is due to the fact that $\frac{|S^n|}{|\mathcal{F}^n|} > \frac{1}{n^d}$. Since each circuit can make at most a polynomial in n number of queries, and since there are a polynomial in n number of circuit evaluations, we can bound the probability of this event occurring to be less than $2^{n/2}$. The argument is made by performing simple counting arguments. ...□

As mentioned earlier, there is one lemma for which the symmetric version of the lemma does not follow immediately from a symmetric version of the argument presented in this chapter. The lemma in question is Lemma 5.7. Note that Lemma 5.7 follows immediately from Lemma 5.8, but observe that in the proof of Lemma 5.8 we need to consider functions of the form $f \circ s_i$, and further observe that an argument similar to the one presented would not work for functions of the form $s_i \circ f$. Therefore, we cannot directly prove a symmetric version of Lemma 5.8 by presenting a proof similar to the one just given for Lemma 5.8, and therefore we cannot conclude a symmetric version of Lemma 5.7. We note that a symmetrical version of Lemma 5.7 may be proven by using an averaging argument, and a slightly stronger version of Lemma 5.7 then is stated. We refer the interested reader to [2] for further details.

5.3 Proof of Lemma 5.2

Assume that there exists a polynomial-sized decision-circuit family $\{C_n\}$, which for some constant $c > 0$ and infinitely many n distinguishes F^n from $\mathcal{F}^n$ with probability at least $\epsilon(n)\delta(2-\delta) + \frac{1}{\log^c n}$.

Lemma 5.9 (Luby-Rackoff) *Either there exists a family of decision-circuits $\{\Lambda_n\}$, where for each n the circuit Λ_n is of size $p_1(\log^c n \cdot c_{\mathsf{G}}(n))s_{\mathsf{F}}(n)$, and for infinitely many n:*

$$\left|\Pr_{\Lambda_n}(\mathsf{G}^n) - \Pr_{\Lambda_n}(\mathcal{F}^n)\right| \geq \delta + \frac{1}{n};$$

or for each n let

$$K_n = \Pr_{C_n}(\mathcal{F}^n) + \delta(1 - \Pr_{C_n}(\mathcal{F}^n)) + \frac{1}{n^{1/3}} \quad \textit{and} \quad L_n = \Pr_{C_n}(\mathcal{F}^n) - \delta \Pr_{C_n}(\mathcal{F}^n) - \frac{1}{n^{1/3}};$$

and let

$$S^n = \left\{ w \in \mathcal{F}^n \,\middle|\, \Pr_{C_n}(\mathsf{G}^n \circ w) \geq K_n \right\} \quad \textit{and} \quad T^n = \left\{ w \in \mathcal{F}^n \,\middle|\, \Pr_{C_n}(\mathsf{G}^n \circ w) \leq L_n \right\}.$$

Then:

$$\Pr_{w \in \mathcal{F}^n}(w \in S^n) \leq \frac{1}{n} \quad \textit{and} \quad \Pr_{w \in \mathcal{F}^n}(w \in T^n) \leq \frac{1}{n},$$

for all sufficiently large n.

Proof(Sketch): If there exists a family of circuits $\{\Lambda_n\}$ with the above mentioned properties then we are finished. Otherwise, no such family of circuits exist, and we need to prove the second case of the lemma.

We introduce the following notation. Let

$$K_n(i) = \Pr_{C_n}(\mathcal{F}^n) + \delta(1 - \Pr_{C_n}(\mathcal{F}^n)) + \frac{1}{n^i} \quad \text{and} \quad L_n(i) = \Pr_{C_n}(\mathcal{F}^n) - \delta \Pr_{C_n}(\mathcal{F}^n) - \frac{1}{n^i};$$

and let

$$S^n(i) = \left\{ w \in \mathcal{F}^n \,\middle|\, \Pr_{C_n}(\mathsf{G}^n \circ w) \geq K_n(i) \right\} \quad \text{and} \quad T^n(i) = \left\{ w \in \mathcal{F}^n \,\middle|\, \Pr_{C_n}(\mathsf{G}^n \circ w) \leq L_n(i) \right\}.$$

Notice that $K_n(\frac{1}{3}) = K_n$, and that $S_n(\frac{1}{3}) = S_n$, and similarly that $L_n(\frac{1}{3}) = L_n$ and that $T_n(\frac{1}{3}) = T_n$.

We will assume for contradiction that for all sufficiently large n the $\Pr_{w^n \in \mathcal{F}^n}(w^n \in S^n(\frac{1}{3})) \geq \frac{1}{n}$. We will show that there exists a family of decision-circuits $\{\widehat{\Lambda}_n\}$, where for each n the circuit $\widehat{\Lambda}_n$ is of size $p_1(\log^c n \cdot c_{\mathsf{G}}(n)) s_{\mathsf{F}}(n)$; and for all sufficiently large n: $\left|\Pr_{\widehat{\Lambda}_n}(\mathsf{G}^n) - \Pr_{\widehat{\Lambda}_n}(\mathcal{F}^n)\right| \geq \delta + \frac{1}{n}$; contradicting the fact that no such family of circuits exist. First we assume that we can randomly and uniformly sample from $S^n(\frac{1}{3})$ (we will deal with the issue of uniform sampling from $S^n(\frac{1}{3})$ later). Then, we can uniformly sample $f_1, ..., f_{n^r} \in (S^n(\frac{1}{3}))^{n^r}$, and construct a probabilistic circuit A_n, which computes $A_n(\nu) = \frac{1}{n^r}\sum_{i=1}^{n^r} C_n(\nu \circ f_i)$. We choose r so that for a randomly chosen function, $\nu \in \mathcal{F}^n$, the probability that $\left|A_n(\nu) - \Pr_{C_n}(\nu \circ S^n(\frac{1}{3}))\right| \geq \frac{1}{n}$ is less than $2^{n/2}$; and for a randomly chosen ν the probability that $|A_n(\nu) - \Pr_{C_n}(\mathcal{F}^n)| \geq \frac{1}{n}$ is less then $2^{n/3}$. Both of these properties are possible due to the Chernoff Bound and Lemma 5.7 respectively.

We will now construct a probabilistic circuit $A'_n(\nu)$ which will accept ν iff $A_n(\nu) > \Pr_{C_n}(\mathcal{F}^n) + \frac{1}{n^{2i}}$. Notice that a random function $\nu \in \mathcal{F}^n$ will be accepted by A'_n with negligible probability. Also notice that $E_{g \in \mathsf{G}^n}[A_n(g)] > K_n(\frac{1}{3})$. Using a simply averaging argument and the two facts just mentioned we can show that at least a fraction $\delta + \frac{1}{n}$ of the $g \in \mathsf{G}^n$ will be accepted by A'_n.

Allowing $\widehat{\Lambda}_n$ to be a derandomized version of A'_n, we have a family of circuits $\{\widehat{\Lambda}_n\}$ for which all sufficiently large n it is the case that $|\Pr_{\Lambda_n}(\mathsf{G}^n) - \Pr_{\Lambda_n}(\mathcal{F}^n)| \geq \delta + \frac{1}{n}$. This provides a contradiction. We can now perform a similar argument by assuming that for all sufficiently large n the $\Pr_{w^n \in \mathcal{F}^n}(w^n \in T^n(\frac{1}{3})) \geq \frac{1}{n}$. We can then derive a similar contradiction. This then proves the lemma.

Unfortunately, in the above argument we assumed that we could sample uniformly from $S^n(\frac{1}{3})$, and in practice this is not possible, so we must show how to get around this. Notice that in the above argument our circuits are significantly smaller than is required by the lemma, and the reason is that many extra gates are needed for properly sampling $S^n(\frac{1}{3})$. We construct a probabilistic circuit $B_n(w) = \frac{1}{n^s}\sum_{i=1}^{n^s} C_n(g_i \circ w)$, where $g_1, ..., g_{n^s}$ are randomly chosen from $(\mathsf{G}^n)^{n^s}$. Using the Chernoff bound, the value s is chosen so that the probability that $|B_n(w) - \Pr_{C_n}(\mathsf{G}^n \circ w)| > \frac{1}{n}$ is negligible. We now create a probabilistic circuit B'_n which accepts a function w iff $B_n(w) \geq K_n(\frac{2}{3})$. This algorithm will accept all but a negligible number of the functions $w \in S^n(\frac{1}{3})$, and it will reject all but a negligible number of functions $w \notin S^n(1)$. We now consider the set $R^n = \{w | w \in S^n(1) \wedge A_n(w) \geq K_n(\frac{2}{3})\}$, and note that since $\frac{|S_n(\frac{1}{3})|}{|\mathcal{F}^n|} \geq \frac{1}{n^d}$, for some d, then it is the case that $\frac{|R^n|}{|\mathcal{F}^n|} \geq \frac{1}{n^{d'}}$, for some d'.

We now simulate the process of choosing random functions in R^n as follows. We run B'_n, but for each function-oracle query, α_i, that the circuit B'_n makes, we substitute the output of the oracle with a random string $r_i \in \{0,1\}^n$. We assume that B'_n has

$m(n)$ oracle gates, and has been constructed so that it never makes the same oracle query twice. Now if B'_n accepts, we know that the set $PF = \bigcup_{i=1}^{m(n)}\{(\alpha_i, r_i)\}$ defines a partial function. All random extensions of PF will be accepted by B'_n. Therefore, except with a negligibly small probability of error, the random extensions of PF are all in R_n. Therefore, we choose a random function in R_n by running B'_n as described above until it accepts. We then save the partial function, and randomly extend it as needed. Since R_n represents a fraction greater than $\frac{1}{n^{1-\lambda}}$ of $\mathcal{F}^n$, for any $0 < \lambda < 1$ and sufficiently large n, we can simulate B'_n a polynomial number of times, and with very high probability we will find a random $w \in R_n$. Therefore, while we cannot randomly sample from $S^n(\frac{1}{3})$, we can sample from $R_n \subset S^n(1)$. Combining this sampling method with small technical modifications to the original argument will allow us to prove the lemma. ...□

Lemma 5.10 *Either there exists a family of decision circuits $\{\Xi_n\}$, where for each n the circuit Ξ_n is of size $c_{\mathsf{H}}(n) + s_{\mathsf{F}}(n)$, and for infinitely many n:*

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

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

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

Proof: If a family of circuit $\{\Xi_n\}$ with the above properties exists, we are done. Otherwise, there exists no such family of circuits. Assume that for the circuit family $\{C_n\}$ there exists a set $\{h^n \in \mathsf{H}^n | n \in \mathbb{N}\}$ such that for all sufficiently large n it is the case that $|\Pr_{C_n}(\mathsf{G}^n \circ h^n) - \Pr_{C_n}(\mathcal{F}^n)| \geq \delta + \frac{1}{n}$. We create the circuit family $\{D_n\}$, where $D_n(\nu) = C_n(\nu \circ h^n)\}$. We note that $\{D_n\}$ is capable of δ distinguishing G from random:

$$\begin{aligned}
\Pr_{D_n}(\mathsf{G}^n) - \Pr_{D_n}(\mathcal{F}^n) &= \Pr_{C_n}(\mathsf{G}^n \circ h^n) - \Pr_{C_n}(\mathcal{F}^n \circ h^n) \\
&= \Pr_{C_n}(\mathsf{G}^n \circ h^n) - \Pr_{C_n}(\mathcal{F}^n) \qquad \text{(Lemma 5.6)} \\
&\geq \delta + \frac{1}{n}
\end{aligned}$$

Letting $D_n = \Xi_n$ we have a constructed a family of circuits $\{\Xi_n\}$. This contradicts the fact that there is no circuit family of size $c_{\mathsf{H}}(n) + s_{\mathsf{F}}(n)$, such that for infinitely many n:

$$\left|\Pr_{\Xi_n}(\mathsf{G}^n) - \Pr_{\Xi_n}(\mathcal{F}^n)\right| \geq \delta + \frac{1}{n},$$

proving the lemma. ...□

5.3.1 Main Argument

Let $\bar{\epsilon}_n = \epsilon(n) + \frac{1}{\log^c n}$ and similarly let $\bar{\delta}_n = \delta + \frac{1}{\log^c n}$. Then the family of decision circuits, $\{C_n\}$, is capable of distinguishing $\mathsf{G}^n \circ \mathsf{H}^n$ from $\mathcal{F}^n$ with probability at least $\bar{\epsilon}_n \bar{\delta}_n (2 - \bar{\delta}_n)$, for infinitely many n, as this value is strictly smaller than $\epsilon\delta(2-\delta) + \frac{1}{\log^c n}$, for sufficiently large n. We assume that there exists no family of decision circuits $\{\Lambda_n\}$, where each circuit is of size $p_1(\log^c n \cdot c_{\mathsf{G}}(n)) s_{\mathsf{F}}(n)$, such that for infinitely many n:

$$\left|\Pr_{\Lambda_n}(\mathsf{G}^n) - \Pr_{\Lambda_n}(\mathcal{F}^n)\right| \geq \delta + \frac{1}{n}.$$

From the above assumption and Lemma 5.9 we know that for

$$K_n = \Pr_{C_n}(\mathcal{F}^n) + \delta(1 - \Pr_{C_n}(\mathcal{F}^n)) + \frac{1}{n^{1/3}} \quad \text{and} \quad S^n = \left\{ w \in \mathcal{F}^n \,\middle|\, \Pr_{C_n}(\mathsf{G}^n \circ w) \geq K_n \right\}, \tag{5.2}$$

that $\Pr_{w \in \mathcal{F}^n}(w \in S^n) \leq \frac{1}{n}$, for all sufficiently large n. Similarly we know that for

$$L_n = \Pr_{C_n}(\mathcal{F}^n) - \delta \Pr_{C_n}(\mathcal{F}^n) - \frac{1}{n^{1/3}} \quad \text{and} \quad T^n = \left\{ w \in \mathcal{F}^n \,\middle|\, \Pr_{C_n}(\mathsf{G}^n \circ w) \leq L_n \right\}, \tag{5.3}$$

that $\Pr_{w \in \mathcal{F}^n}(w \in T^n) \leq \frac{1}{n}$, for all sufficiently large n.

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

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

From the above assumption and Lemma 5.10, we know that for all sufficiently large n and all $h^n \in \mathsf{H}^n$:

$$\left|\Pr_{C_n}(\mathsf{G} \circ h^n) - \Pr_{C_n}(\mathcal{F}^n)\right| < \delta + \frac{1}{n}. \tag{5.4}$$

We will construct a family of decision circuits $\{\Upsilon_n\}$, where each circuit Υ_n is of size $p_2(\log^c n \cdot c_{\mathsf{G}}(n))s_{\mathsf{F}}(n)$, such that for infinitely many n:

$$\left| \Pr_{\Upsilon_n}(\mathsf{H}^n) - \Pr_{\Upsilon_n}(\mathcal{F}^n) \right| \geq \epsilon(n) + \frac{1}{\log^{2c} n},$$

proving the lemma. In the sequel we show how to construct Υ_n for each n such that both $|\Pr_{C_n}(\mathsf{F}^n) - \Pr_{C_n}(\mathcal{F}^n)| \geq \epsilon(n)\delta(2 - \max\{\epsilon(n), \delta\}) + \frac{1}{n^c}$, and n is sufficiently large for all inequalities in the following proof to hold true.

We first give an overview of the proof. We will construct a circuit B_n (via an intermediate circuit A_n) which, given as input a function w, almost surely approximates the value $\Pr_{C_n}(\mathsf{G}^n \circ w)$. From Equations 5.2 and 5.3, we know that for almost all random functions $w \in \mathcal{F}^n$ that $\Pr_{C_n}(\mathsf{G}^n \circ w)$ will be bounded from above by the value K_n, and below by the value L_n. We break the proof into two cases.

In the first case, if for a large fraction of the $h \in \mathsf{H}_n$ (a fraction greater than $\epsilon(n) + \frac{1}{\log^c n}$) it is the case that $\Pr_{C_n}(\mathsf{G}^n \circ h)$ falls outside of the range $[L_n, K_n]$, then we can distinguish between H^n and $\mathcal{F}^n$ by computing $B(w)$, and accepting if $B(w)$ is greater than K_n or less than L_n. Because the sets S^n and T^n contain all of the functions w in $\mathcal{F}^n$, for which $\Pr_{C_n}(\mathsf{G}^n \circ w)$ is less than or greater than L_n and K_n respectively, and since $\frac{|S^n|}{|\mathcal{F}^n|} < \frac{1}{n}$ and $\frac{|T^n|}{|\mathcal{F}^n|} < \frac{1}{n}$; we know that about a fraction $\frac{2}{n}$ of the w will be accepted by $B(n)$.

In the second case there is not a large enough fraction of $h \in \mathsf{G}^n$ for which $\Pr_{C_n}(\mathsf{G}^n \circ h)$ falls outside of the range $[L_n, K_n]$. In this case, to distinguish H^n from $\mathcal{F}^n$ we use a technique of Levin's, and construct a circuit D_n which on input w tosses a biased coin which is heads with probability $B_n(w)$, and tails otherwise. The circuit D_n accepts the input w if the result of the flip is heads. We will now present the technical details of the proof which was just described.

We construct a probabilistic circuit A_n such that

$$A_n(w) = \frac{1}{\log^b n} \sum_{i=1}^{\log^b n} C_n(g_i \circ w),$$

where $g_1, ..., g_{n^r}$ are randomly chosen from G^n. Using the Chernoff Bound b is chosen large enough such that:

$$\Pr_{w \in \mathcal{F}^n}\left[\left|A_n(w) - \Pr_{C_n}(\mathsf{G} \circ w)\right| \geq \frac{1}{\log^{4c} n}\right] \leq \frac{1}{n^2},$$

and

$$\Pr_{h^n \in \mathsf{H}^n}\left[\left|A_n(h^n) - \Pr_{C_n}(\mathsf{G} \circ h^n)\right| \geq \frac{1}{log^{4c} n}\right] \leq \frac{1}{n^2}.$$

By Lemma 5.5 we derandomize A_n to get B_n so that for all but $\frac{1}{n}$ of the $w \in \mathcal{F}^n$:

$$\left|B_n(w) - \Pr_{C_n}(\mathsf{G}^n \circ w)\right| < \frac{1}{\log^{4c} n}, \tag{5.5}$$

and for all but $\frac{1}{n}$ of the $h^n \in \mathsf{H}^n$:

$$\left|B_n(h^n) - \Pr_{C_n}(\mathsf{G}^n \circ h^n)\right| < \frac{1}{\log^{4c} n}. \tag{5.6}$$

We now break the proof into the two cases mentioned earlier. The first case covers the situation in which there is a clear separation between the value output by B_n on functions from $\mathcal{F}^n$ versus functions from H^n. The second case handles the random coin flipping.

Let $K'_n = \Pr_{C_n}(\mathcal{F}^n) + \bar{\delta}_n(1 - \Pr_{C_n}(\mathcal{F}^n))$, and let $L'_n = \Pr_{C_n}(\mathcal{F}^n) - \bar{\delta}_n \Pr_{C_n}(\mathcal{F}^n)$. We note that in the following proof that the variable K_n seems to be used many times, but there appears to be little justification for the variable L'_n. We note that it replaces the use of K'_n in the symmetric version of this proof, and therefore it is included in this version of the proof as well.

Case 1: $\Pr_{h^n \in \mathsf{H}^n}[B_n(h^n) \geq K'_n] \geq \bar{\epsilon}_n = \epsilon(n) + \frac{1}{\log^c n}$

We create the decision circuit $\bar{B}_n(w)$ which accepts w iff $B_n(w) \geq K'_n$.

$$\begin{aligned}
\left|\Pr_{\bar{B}_n}(\mathsf{H}^n) - \Pr_{\bar{B}_n}(\mathcal{F}^n)\right| &\geq \epsilon(n) + \frac{1}{\log^c n} - \Pr_{\bar{B}_n}(\mathcal{F}^n) \\
&\geq \epsilon(n) + \frac{1}{\log^c n} - \frac{1}{n} - \frac{1}{n} \quad \text{(Equation 5.2 \& sampling error)} \\
&\geq \epsilon(n) + \frac{1}{\log^{c+1} n}
\end{aligned}$$

Notice that the second inequality follows from constraints on the probability that a random function $w \in \mathcal{F}^n$ is in the set S^n defined in equation 5.2. This is because, for all sufficiently large n, K'_n is greater than the value of K_n. As previously seen, the probability that a random $w \in \mathcal{F}^n$ has the property that $\Pr_{C_n}(\mathsf{G}^n \circ w) > K_n$, is the probability that w is contained in the set S^n. But, we know that the probability of a random $w \in \mathcal{F}^n$ being in the set S^n is less than $\frac{1}{n}$. Clearly it follows that the probability that a random $w \in \mathcal{F}^n$ has the property that $\Pr_{C_n}(\mathsf{G}^n \circ w) > K'_n$ is smaller than $\frac{1}{n}$, for the set of functions with this property is necessarily a subset of S^n. Finally, since $B_n(w)$ approximates $\Pr_{C_n}(\mathsf{G}^n \circ w)$ to within a value of $\frac{1}{\log^{4c} n}$, for all but $\frac{1}{n}$ of the $w \in \mathcal{F}^n$, and since it still holds that for all sufficiently large n that $K'_n - \frac{1}{\log^{4c} n}$ is greater than K_n, there can only be a fraction $\frac{2}{n}$ of the $w \in \mathcal{F}^n$ for which $B'(w)$ accepts. The first $\frac{1}{n}$ of these functions are the ones in S^n. The last $\frac{1}{n}$ of these functions are the functions w in which $B(w)$'s approximation of $\Pr_{C_n}(\mathsf{G}^n \circ w)$ is off by more than a factor of $\frac{1}{\log^{4c} n}$.

We let $\Upsilon_n = \bar{B}_n$, and we are finished Case 1.

Case 2: $\Pr_{h^n \in \mathsf{H}^n}[B_n(h^n) \geq K'_n] < \bar{\epsilon}_n = \epsilon(n) + \frac{1}{\log^c n}$

Let $q(w) = B_n(w)$ and let

$$q'(w) = \begin{cases} K'_n & \text{if } K'_n < B_n(w); \\ q(w) & \text{if } L'_n \leq B_n(w) \leq K'_n; \\ L'_n & \text{if } B_n(w) < L'_n. \end{cases}$$

Let

$$p(w) = \frac{q(w) - \Pr_{C_n}(\mathcal{F}^n)}{\delta} + \Pr_{C_n}(\mathcal{F}^n) \quad \text{and let} \quad p'(w) = \frac{q'(w) - \Pr_{C_n}(\mathcal{F}^n)}{\bar{\delta}_n} + \Pr_{C_n}(\mathcal{F}^n).$$

We use B_n to create a probabilistic decision circuit D_n, which accepts w with probability $p'(w)$. We will show that D_n distinguishes between H^n and $\mathcal{F}^n$ with probability at least $\epsilon(n) + \frac{1}{\log^{2c} n}$, proving the theorem. However, we first make some observations about the circuit B_n which we will use later in the proof. Notice that,

$$\left| E_{w \in \mathcal{F}^n}(B_n(w)) - \Pr_{C_n}(\mathsf{G}^n \circ \mathcal{F}^n) \right| \leq \frac{1}{\log^{4c} n} + \frac{1}{n} \quad \text{and}$$

$$\left| E_{h^n \in \mathsf{H}^n}(B_n(h^n)) - \Pr_{C_n}(\mathsf{G}^n \circ \mathsf{H}^n) \right| \leq \frac{1}{\log^{4c} n} + \frac{1}{n}.$$

These facts follow from equations (5.5) and (5.6) respectively. We use the first inequality to get an upper bound on $\Pr_{D_n}(\mathcal{F}^n)$, and the second to get a lower bound on $\Pr_{D_n}(\mathsf{H}^n)$.

We observe that $\Pr_{D_n}(\mathcal{F}^n) = E_{w \in \mathcal{F}^n}(p'(w))$. We first produce an upper bound on the expectation of $p(w)$, and then we compensate for the possible discrepancy between it and $p'(w)$.

$$\begin{aligned}
E_{w \in \mathcal{F}^n}(p(w)) &= \frac{E_{w \in \mathcal{F}^n}(q(w)) - \Pr_{C_n}(\mathcal{F}^n)}{\bar{\delta}_n} + \Pr_{C_n}(\mathcal{F}^n) \\
&= \frac{E_{w \in \mathcal{F}^n}(B_n(w)) - \Pr_{C_n}(\mathcal{F}^n)}{\bar{\delta}_n} + \Pr_{C_n}(\mathcal{F}^n) \\
&\leq \frac{\Pr_{C_n}(\mathsf{G}^n \circ \mathcal{F}^n) + \frac{1}{\log^{4c} n} + \frac{1}{n} - \Pr_{C_n}(\mathcal{F}^n)}{\bar{\delta}_n} + \Pr_{C_n}(\mathcal{F}^n) \\
&= \frac{\Pr_{C_n}(\mathcal{F}^n) + \frac{1}{\log^{4c} n} + \frac{1}{n} - \Pr_{C_n}(\mathcal{F}^n)}{\bar{\delta}_n} + \Pr_{C_n}(\mathcal{F}^n) \\
&= \frac{\frac{1}{\log^{4c} n} + \frac{1}{n}}{\bar{\delta}_n} + \Pr_{C_n}(\mathcal{F}^n) \\
&\leq \Pr_{C_n}(\mathcal{F}^n) + \frac{1}{\log^{3c} n}
\end{aligned}$$

We need to take into account the possibility that $p(w) < 0$, for this can never happen with $p'(w)$. This is because $q(w)$ may be smaller than L'_n whereas this cannot be the case with $q'(w)$. Fortunately, by Equation 5.3, we know that the probability that $q(w) < L'_n$ is non-significant.

$$\begin{aligned}
\Pr_{B_n}(\mathcal{F}^n) &= E_{w \in \mathcal{F}^n}(p'(w)) \\
&\leq E_{w \in \mathcal{F}^n}(p(w)) + \frac{1}{n} + \frac{1}{n} \qquad \text{(Equation 5.3 and sampling error)} \\
&\leq \Pr_{C_n}(\mathcal{F}^n) + \frac{1}{\log^{3c} n} + \frac{1}{n} + \frac{1}{n} \\
&\leq \Pr_{C_n}(\mathcal{F}^n) + \frac{1}{\log^{2c} n}
\end{aligned}$$

We now give a lower bound for $Pr_{D_n}(\mathsf{H}^n) = E_{h^n \in \mathsf{H}^n}(p'(h^n))$. This is done by showing a lower bound for $E_{h^n \in \mathsf{H}^n}(p(h^n))$, and then compensating for the possible difference between $p(h^n)$ and $p'(h^n)$.

$$\begin{aligned}
E_{h^n \in \mathsf{H}^n}(p(h^n)) &= \frac{E_{h^n \in \mathsf{H}^n}(q(h^n)) - \Pr_{C_n}(\mathcal{F}^n)}{\bar{\delta}_n} + \Pr_{C_n}(\mathcal{F}^n) \\
&= \frac{E_{h^n \in \mathsf{H}^n}(B_n(h^n)) - \Pr_{C_n}(\mathcal{F}^n)}{\bar{\delta}_n} + \Pr_{C_n}(\mathcal{F}^n) \\
&\geq \frac{\Pr_{C_n}(\mathsf{G}^n \circ \mathsf{H}^n) - \frac{1}{\log^{4c} n} - \frac{1}{n} - \Pr_{C_n}(\mathcal{F}^n)}{\bar{\delta}_n} + \Pr_{C_n}(\mathcal{F}^n) \\
&\geq \frac{\bar{\epsilon}_n \bar{\delta}_n (2 - \bar{\delta}_n) - \frac{1}{\log^{4c} n} - \frac{1}{n}}{\bar{\delta}_n} + \Pr_{C_n}(\mathcal{F}^n) \\
&\geq \bar{\epsilon}_n (2 - \bar{\delta}_n) - \frac{1}{\bar{\delta}_n \log^{4c} n} - \frac{1}{\bar{\delta}_n n} + \Pr_{C_n}(\mathcal{F}^n) \\
&\geq \bar{\epsilon}_n (2 - \bar{\delta}_n) - \frac{1}{\log^{3c} n} + \Pr_{C_n}(\mathcal{F}^n)
\end{aligned}$$

We compensate for the $h^n \in \mathsf{H}^n$ such that $p(h^n) > 1$. This occurs when $q(h^n) > K'$, and thus by virtue of being in case 2, this can only occur for a fraction $\bar{\epsilon}_n = \epsilon(n) + \frac{1}{\log^c n}$ of the $h^n \in \mathsf{H}^n$.

By Equation 5.4 we know that $\forall h_k^n \in \mathsf{H}^n \; |\Pr_{C_n}(\mathsf{G} \circ h_k^n) - \Pr_{C_n}(\mathcal{F}^n)| < \delta + \frac{1}{n}$. Note that $\Pr_{C_n}(\mathsf{G}^n \circ h_k^n)$ is within $\frac{1}{\log^{4c} n}$ of $B_n(h^n) = q(h^n)$, for all but $\frac{1}{n}$ of the $h^n \in \mathsf{H}^n$. Further, the $\Pr_{C_n}(\mathsf{G}^n \circ h_k^n)$ is at most one. It follows that $|q(h^n) - \Pr_{C_n}(\mathcal{F}^n)| < \delta + \frac{1}{\log^{4c} n} + \frac{1}{n}$, for most $h^n \in \mathsf{H}^n$. We conclude that *for all but $\frac{1}{n}$ of the $h^n \in \mathsf{H}^n$:*

$$q(h) < \min\left\{ \left(1 + \frac{1}{\log^{4c} n} + \frac{1}{n}\right), \left(\Pr_{C_n}(\mathcal{F}^n) + \delta + \frac{1}{\log^{4c} n} + \frac{1}{n}\right) \right\}.$$

We show that in both of the above cases the distance between $q(h^n)$ and $q'(h^n) = K'_n$ is less than $\bar{\delta}_n(1-\delta)$.

Sub Case 1) $\Pr_{C_n}(\mathcal{F}^n) \leq (1-\delta)$

It must be the case that $q(h^n) < \Pr_{C_n}(\mathcal{F}^n) + \delta + \frac{1}{\log^{4c} n} + \frac{1}{n}$ and $q'(h^n) = K'_n = \Pr_{C_n}(\mathcal{F}^n) + \bar{\delta}_n(1 - \Pr_{C_n}(\mathcal{F}^n))$.The result follows:

$$q(h^n) - q'(h^n) \;<\; \left(\Pr_{C_n}(\mathcal{F}^n) + \delta + \frac{1}{\log^{4c} n} + \frac{1}{n}\right) - \left(\Pr_{C_n}(\mathcal{F}^n) + \bar{\delta}_n(1 - \Pr_{C_n}(\mathcal{F}^n)\right)$$

$$
\begin{aligned}
&= \left(\delta + \frac{1}{\log^{4c} n} + \frac{1}{n}\right) - \left(\delta + \frac{1}{\log^{c} n}\right)\left(1 - \Pr_{C_n}(\mathcal{F}^n)\right) \\
&= \left(\delta + \frac{1}{\log^{4c} n} + \frac{1}{n}\right) - \delta + \delta \Pr_{C_n}(\mathcal{F}^n) - \frac{1}{\log^{c} n} + \frac{1}{\log^{c} n}\Pr_{C_n}(\mathcal{F}^n) \\
&= \left(\frac{1}{\log^{4c} n} + \frac{1}{n}\right) + \delta \Pr_{C_n}(\mathcal{F}^n) - \frac{1}{\log^{c} n} + \frac{1}{\log^{c} n}\Pr_{C_n}(\mathcal{F}^n) \\
&< \delta \Pr_{C_n}(\mathcal{F}^n) + \frac{1}{\log^{c} n}\Pr_{C_n}(\mathcal{F}^n) \\
&= \bar{\delta}_n \Pr_{C_n}(\mathcal{F}^n) \\
&< \bar{\delta}_n(1-\delta).
\end{aligned}
$$

Sub Case 2) $\Pr_{C_n}(\mathcal{F}^n) > (1-\delta)$

In this case there must exist a u such that $0 < u \leq \delta$ where $\Pr_{C_n}(\mathcal{F}^n) = 1 - \delta + u$. The result follows:

$$
\begin{aligned}
q(h) - q'(h) &< 1 + \frac{1}{\log^{4c} n} + \frac{1}{n} - K'_n \\
&= \Pr_{C_n}(\mathcal{F}^n) + \delta - u + \frac{1}{\log^{4c} n} + \frac{1}{n} - K'_n \\
&= \Pr_{C_n}(\mathcal{F}^n) + \delta - u + \frac{1}{\log^{4c} n} + \frac{1}{n} - \left(\Pr_{C_n}(\mathcal{F}^n) + \bar{\delta}_n(1 - \Pr_{C_n}(\mathcal{F}^n)\right) \\
&= -u + \frac{1}{\log^{4c} n} + \frac{1}{n} - \left(\bar{\delta}_n(1 - \Pr_{C_n}(\mathcal{F}^n)\right) \\
&= -u + \frac{1}{\log^{4c} n} + \frac{1}{n} - \left(\bar{\delta}_n - \bar{\delta}_n \Pr_{C_n}(\mathcal{F}^n)\right) \\
&= -u + \frac{1}{\log^{4c} n} + \frac{1}{n} - \left(\delta + \frac{1}{\log^{c} n} - \bar{\delta}_n \Pr_{C_n}(\mathcal{F}^n)\right) \\
&= -u - \delta + \bar{\delta}_n \Pr_{C_n}(\mathcal{F}^n) \\
&= -u - \delta + \bar{\delta}_n(1 - \delta + u) \\
&= -u - \delta + \bar{\delta}_n(1-\delta) + \bar{\delta}_n u \\
&= \bar{\delta}_n(1-\delta) + \bar{\delta}_n u - u - \delta \\
&\leq \bar{\delta}_n(1-\delta).
\end{aligned}
$$

Using the bound on the distance between $p(h^n)$ and $p'(h^n)$ derived above and applying it to the previous lower bound for $\Pr_{D_n}(\mathsf{H}^n)$ we get that for all but $\frac{1}{n}$ of $h^n \in \mathsf{H}^n$:

$$\begin{aligned}
\Pr_{D_n}(\mathsf{H}^n) &= E_{h^n\in\mathsf{H}^n}(p'(h^n)) \\
&\geq E_{h^n\in\mathsf{H}^n}(p(h^n)) - \left(\epsilon(n) + \frac{1}{\log^c n}\right)(1-\delta) \\
&= E_{h^n\in\mathsf{H}^n}(p(h^n)) - (\bar{\epsilon}_n)(1-\delta)
\end{aligned}$$

For the $\frac{1}{n}$ of the $h^n \in \mathsf{H}^n$ that are not within the distance given by the Chernoff bound, we know that they differ by at most 1 from the value being estimated, and therefore can conclude:

$$\begin{aligned}
\Pr_{D_n}(\mathsf{H}^n) &= E_{h^n\in\mathsf{H}^n}(p'(h^n)) \\
&\geq E_{h^n\in\mathsf{H}^n}(p(h^n)) - [\bar{\epsilon}_n][(1-\delta)] - \frac{1}{n} \\
&= [\bar{\epsilon}_n(2-\bar{\delta}_n) - \frac{1}{\log^{3c} n} + \Pr_{C_n}(\mathcal{F}^n)] - [\bar{\epsilon}_n][(1-\delta)] - \frac{1}{n} \\
&= [\bar{\epsilon}_n(2-\bar{\delta}_n - 1 + \delta) - \frac{1}{\log^{2c} n} + \Pr_{C_n}(\mathcal{F}^n)] \\
&= [\bar{\epsilon}_n(1-\bar{\delta}_n + \delta) - \frac{1}{\log^{2c} n} + \Pr_{C_n}(\mathcal{F}^n)] \\
&= [\bar{\epsilon}_n(1-\delta - \frac{1}{\log^c n} + \delta) - \frac{1}{\log^{2c} n} + \Pr_{C_n}(\mathcal{F}^n)] \\
&= \bar{\epsilon}_n(1 - \frac{1}{\log^c n}) - \frac{1}{\log^{2c} n} + \Pr_{C_n}(\mathcal{F}^n).
\end{aligned}$$

By taking the difference between the lower bound on the acceptance rate of D_n on H^n and the upper bound on the acceptance rate of D_n on $\mathcal{F}^n$ we achieve the proper distinguishing probability.

$$
\begin{aligned}
\Pr_{D_n}(\mathsf{H}^n) - \Pr_{D_n}(\mathcal{F}^n) &\geq [\bar{\epsilon}_n(1 - \frac{1}{\log^c n}) - \frac{1}{\log^{2c} n} + \Pr_{C_n}(\mathcal{F}^n)] - [\Pr_{C_n}(\mathcal{F}^n) + \frac{1}{\log^{2c} n}] \\
&\geq \bar{\epsilon}_n(1 - \frac{1}{\log^c n}) \\
&= (\epsilon(n) + \frac{1}{\log^c n})(1 - \frac{1}{\log^c n}) \\
&= (\epsilon(n) - \frac{\epsilon(n)}{\log^c n} + \frac{1}{\log^c n} - \frac{1}{\log^{2c} n}) \\
&\geq \epsilon(n) + \frac{1}{\log^{2c} n}
\end{aligned}
$$

All that remains is to derandomize D_n. In this case we let Υ_n be the derandomized version of D_n. We have now constructed the family of circuits $\{\Upsilon_n\}$ which was required to prove the lemma. ...□

5.4 Towards Complete Security

Our goal is to show that by composing enough partially secure PRPGs together we can construct a PRPG which is completely secure. So far our efforts along these lines have failed. We will now discuss some possible future research directions which will allow us to attain our goal.

Observe that although the Isolation Lemma is tight, we have been unable to find a set of three permutation generators which when composed together are as insecure as is permitted by the direct application of the Isolation Lemma. In particular, in our previous example we showed that there are generators G which are $1 - \delta$ secure, and generators H which are $1 - \epsilon$ secure, but whose composition $\mathsf{G} \circ \mathsf{H}$ is $\epsilon\delta(2 - \max\{\epsilon, \delta\})$ secure. However, a crucial requirement of this example is that a fraction δ of the keys in G correspond to the identity permutation, and a fraction ϵ of the keys in H correspond to the identity permutation. However, only a fraction $\epsilon\delta$ of the keys for $\mathsf{G} \circ \mathsf{H}$ correspond to the identity permutation, whereas its security is $\epsilon\delta(2-\max\{\epsilon,\delta\})$. This fact prevents us from composing $\mathsf{G}\circ\mathsf{H}$ with a $1-\gamma$ secure generator F to produce a generator $\mathsf{F}\circ(\mathsf{G}\circ\mathsf{H})$ which is $\epsilon\delta\gamma(2-\max\{\epsilon,\delta\})(2-\max\{\epsilon,\delta,\gamma\})$ secure. The observation we make is that the fraction of keys for $\mathsf{G}\circ\mathsf{H}$ which correspond to permutations which are easily distinguishable from random functions is much smaller than the resulting security of $\mathsf{G}\circ\mathsf{H}$. Further, it is this lack of "special", and therefore easily identifiable, permutations in $\mathsf{G}\circ\mathsf{H}$ which prevents us from composing the generator with a third, and achieving the security bound specified by the Isolation Lemma.

We conjecture that the security of the composition of three or more partially secure PRPGs is stronger than the bound which results from multiple applications of the Isolation Lemma. However, we suspect that the current notion of security may be too coarse to prove such a statement. Therefore, we propose a finer notion of security.

Definition 5.1 (Weak Security) *We say that a PRPG (PRFG)* G *is* $1-\delta(n)$ *weak-secure if for all adversaries* A^f *which have the property that for all* $c>0$*, and for all sufficiently large* n*:*

$$\Pr_{f\in\mathcal{F}^n}[A^f=1]\le\frac{1}{n^c};$$

then it is the case that for all sufficiently large n *and all* $c>0$*:*

$$\left|\Pr_{f\in\mathcal{F}^n}[A^f=1]-\Pr_{g\in\mathsf{G}^n}[A^g=1]\right|\le\delta(n)+\frac{1}{n^c}.$$

We note that by some minor modifications to the arguments presented in this chapter, it is easy to show that the composition of a $1-\delta$ weakly secure PRPG and a $1-\epsilon$ weakly secure PRPG is $1-\delta\epsilon$ weakly-secure (This follows from the fact that sub case 2 of case 2 in section 5.3.1 disappears). This shows that there is no way of increasing the fraction of permutations which are easily identifiable from random in the composition of two partially secure PRPGs. We suspect that further work in this direction may point the way to developing a theorem which shows that a polynomial in n number of compositions of a $1-\delta$ partially secure PRPG results in a 1 secure PRPG.

Chapter 6

The Luby-Rackoff Operator

In Chapter 5 we showed how to take partially secure PRPGs and compose them to produce a PRPG which is more secure than any of the constituent PRPGs. In this chapter we will show that an analog is possible for PRFGs. As was demonstrated in Chapter 4, this cannot be done directly by composing PRFGs, and so we introduce a new operator $\Box$ (read as box). We will show that the $\Box$ operator acting on functions has some properties in common with the composition operator acting on permutations.

Given two functions f_1 and f_2 such that $f_i : \{0,1\}^n \to \{0,1\}^n$, for $i \in \{1,2\}$, we define the operator $\Box$ as:

$$(f_1 \Box f_2)(x) = f_1(f_1(x) \oplus f_2(x)) \oplus f_2(f_1(x) \oplus f_2(x)).$$

The $\Box$ operator was originally defined in some working notes by Luby and Rackoff [13] in which an incomplete argument is given to show that two partially secure PRFGs can be combined by the $\Box$ operator, so that the resulting PRFG is more secure than either of its constituents. We will now complete the argument, and show how these results relate to the arguments presented in Chapter 5.

6.1 Two Technical Lemmas

In Chapter 5 we presented two Isolation Lemmas which proved that the composition of two partially secure PRPGs resulted in a third which was more secure than either of the constituent generators. However, in both proofs the only arguments which made use of specific properties of the composition operator or the permutation property of the PRPGs were contained in Lemmas 5.6 and 5.7. Therefore, if we can prove corresponding lemmas for the $\Box$ operator and PRFGs, then equivalent Isolation Lemmas will hold for the $\Box$ operator acting on PRFGs.

Before proving these correspondences we standardize the circuits that will be used in the remainder of this chapter. Observe that whenever we have a circuit C_n which has oracle-gates which correspond to a function f, it can be easily modified, by adding at most a polynomial number of gates and connections between gates, to render a new circuit $\widehat{C_n}$, where $\widehat{C_n}$ never repeats queries to f on the same input. Further, any polynomial sized family of circuits $\{C_n\}$, in which each circuit C_n has at most $m(n)$ oracle gates, can be easily be modified to a equivalent family of circuits $\{\bar{C_n}\}$, where each circuit $\bar{C_n}$ has exactly $m(n)$ oracle gates. We shall assume that all circuits and circuit families in this chapter are of the modified forms described above, unless specifically mentioned otherwise.

We now state the first property of the $\Box$ operator, which we will use to develop one of the above mentioned correspondences.

Lemma 6.1 *Given any family of decision-circuits $\{C_n\}$, for each $f \in \mathcal{F}^n$,*

$$|\Pr_{C_n}(f\Box\mathcal{F}^n) - \Pr_{C_n}(\mathcal{F}^n)| \leq \frac{2m^2(n)}{2^n},$$

where $m(n)$ is the number oracle gates in the circuit C_n.

Intuitively the lemma shows that the $\Box$ operator maintains the security of its strongest operand, for there is a trivial circuit which distinguishes f, but by the above lemma there can be no such circuit for $f\Box\mathcal{F}^n$. For completeness we give the proof of Lemma 6.1 originally presented by Luby and Rackoff in [13].

Proof of Lemma 6.1: We will perform two experiments which will result in two random variables R_1 and R_2. We will show that $E[R_1] = \Pr_{C_n}(\mathcal{F}^n)$, and then show that $E[R_2]$ differs from $\Pr_{C_n}(f\Box\mathcal{F}^n)$ by at most a negligible amount. Finally, we will observe that $R_1 = R_2$, proving the result.

In the first experiment we let $X_1, .., X_{m(n)}$ be i.i.d.r.v. from the uniform distribution over $\{0,1\}^n$. Let $g_1, ..., g_{m(n)}$ be the oracle gates in C_n. We define R_1 to be the output of the circuit C_n when we replace the output of gate g_i with the value X_i. Let α_i be the input to oracle gate g_i. Note that for i, such that $1 < i \leq m(n)$, that α_i is completely determined by C_n and $X_1, ..., X_{i-1}$. This taken in conjunction with the fact that for $i \neq j$ that $\alpha_i \neq \alpha_j$ and by the construction of C_n, we observe that $\Pr_{C_n}(\mathcal{F}^n) = E[R_1]$.

For the second experiment, observe that $\Pr_{C_n}(f\Box\mathcal{F}^n) = \Pr_{f'\in\mathcal{F}^n}[C_n(f' \circ f') = 1]$. This follows from two observations: first $(f_1\Box f_2)(x) = (f_1 \oplus f_2) \circ (f_1 \oplus f_2)(x)$; secondly, for any $\hat{f} \in \mathcal{F}^n$ the set $\hat{f} \oplus \mathcal{F}^n$ is isomorphic to $\mathcal{F}^n$. We construct the circuit $\widehat{C}_n$ by replacing each oracle-gate g_i in the circuit C_n with two oracle-gates h_i and h_i'. We feed the original input of g_i into h_i; feed the output of h_i into h_i'; and let the output of h_i' replace the original output of g_i. We now let $X_1, .., X_{m(n)}$ and $Y_1, ..., Y_{m(n)}$ be i.i.d.r.v. from the uniform distribution over $\{0,1\}^n$. We define R_2 to be the the output of $\widehat{C}_n$ when we replace the output of gate h_i with Y_i, and the output of h_i' with X_i, for $1 \leq i \leq m(n)$. Let α_i be the input to gate h_i, and notice that α_i is completely determined by $\widehat{C}_n$ and $X_1, ..., X_{i-1}$. Clearly $E[R_1] = E[R_2]$. But we will show that $E[R_2]$ is negligibly close to $\Pr_{C_n}(f\Box\mathcal{F}^n)$. We consider the set

$$\text{good} = \left\{X_1, ..., X_{m(n)}, Y_1, ..., Y_{m(n)} \middle| \text{ the relation } \{(\alpha_i, Y_i)\} \cup \{(Y_i, X_i)\} \text{ is one-to-one}\right\}.$$

The set contains all of the choices of $X_1, ..., X_{m(n)}, Y_1, ..., Y_{m(n)}$ which define a partial function in the above experiment. We then define the set *bad* to be the complement of the set *good*. Notice that:

$$\begin{aligned}
\Pr_{C_n}(f\Box\mathcal{F}^n) &= \Pr_{f'\in\mathcal{F}^n}\left[C_n(f'\circ f')=1\right] \\
&= \Pr_{f'\in\mathcal{F}^n}\left[\widehat{C}_n(f')=1\right] \\
&= E_{f'\in\mathcal{F}^n}\left[\widehat{C}_n(f')\right] \\
&= E_{t=\left(X_1,..,X_{m(n)},Y_1,...,Y_{m(n)}\right)\in\{0,1\}^{n\times 2m(n)}}\left[R_2|t\in\text{good}\right]
\end{aligned}$$

Finally, observe that a tuple $\left(X_1,...,X_{m(n)},Y_1,...,Y_{m(n)}\right)$ is definitely in good if $\alpha_1,..,\alpha_{m(h)},Y_1,...,Y_{m(n)}$ are all distinct. But by the construction of $\widehat{C}_n$, $\alpha_i\neq\alpha_j$ for $i\neq j$, and thus the probability that these values are not all distinct is bounded above by $\frac{2m^2(n)}{2^n}$. Observing that:

$$\begin{aligned}
\Pr_{C_n}(\mathcal{F}^n) &= E[R_2] \\
&= E[R_2|\text{good}]\Pr[\text{good}]+E[R_2|\text{bad}]\Pr[\text{bad}] \\
&\leq \Pr_{C_n}(f\Box\mathcal{F}^n)\Pr[\text{good}]+E[R_2|\text{bad}]\frac{2m^2}{2^n} \\
&\leq \Pr_{C_n}(f\Box\mathcal{F}^n)+\frac{2m^2(n)}{2^n},
\end{aligned}$$

the result follows. ...□

Corollary 6.1 *For each n let $S^n\subset\mathcal{F}^n$, and let $\{C_n\}$ be a family of circuits, then*

$$|\Pr_{C_n}(S^n\Box\mathcal{F}^n)-\Pr_{C_n}(\mathcal{F}^n)|<\frac{2m^2(n)}{2^n},$$

where $m(n)$ is the number oracle gates in the circuit C_n.

Notice the similarities between Corollary 6.1 and Lemma 5.6. In effect, except for a negligibly small chance of error, we have proven the equivalent of Lemma 5.6 except that the composition operator and a set of permutations has been replaced by the □ operator and a set of functions, respectively. However, because the probability of error is negligible it does not have an effect on arguments made in the proofs of the Isolation Lemmas. Therefore, we can intuitively treat $\Pr_{C_n}(S^n\Box\mathcal{F}^n)=\Pr_{C_n}(\mathcal{F}^n)$.

We now present the second property of the $\Box$ operator. As we used Lemma 5.8 to prove Lemma 5.7, we will similarly use a lemma, which corresponds to Lemma 5.8, to help prove a Lemma which corresponds to Lemma 5.7. In order to prove the lemma which corresponds to Lemma 5.8, we first need to prove the lemma given below.

Lemma 6.2 *For every n let $\widehat{S}^n \subset \mathcal{F}^n$ be a set such that $\frac{|\widehat{S}^n|}{|\mathcal{F}^n|} \geq \frac{1}{n^c}$, for some constant $c > 0$. Let $\{B_n\}$ be an infinite family of polynomial in n sized oracle circuits. For each n, let B_n have exactly $m(n)$ oracle gates. Then for all k and all sufficiently large n*

$$\Pr_{\hat{s} \in \widehat{S}^n}[B_n(\hat{s} \circ \hat{s}) = (\eta_k, \hat{s}(\eta_k))] \leq \frac{n^c \cdot m(n)^3}{2^n},$$

where η_k is the k^{th} query made to a function oracle gate by B_n when given $\hat{s} \circ \hat{s}$ as an oracle.

Proof: We first give a bound on the probability when $\hat{s}$ is chosen uniformly at random from the set $\mathcal{F}^n$. Later, we use this bound to derive one in which $\hat{s}$ is considered to be chosen from the set $\widehat{S}^n$.

Let B_n be a polynomial sized oracle circuit which has $m(n)$ oracle gates. Further, suppose that B_n never performs the same query twice. Let $\Gamma_1, ..., \Gamma_{m(n)}$ be the oracle gates of B_n. Each gate Γ_k is supposed to represent the function $\hat{s} \circ \hat{s}$. Therefore, we replace each gate Γ_k with the gates Γ_k^1 and Γ_k^2, where these gates each represent the function $\hat{s}$. This is done by taking the input to Γ_k and redirecting it to Γ_k^1; taking the output of Γ_k^1 and making it the input of Γ_k^2; and replacing the output of Γ_k with the output of Γ_k^2.

We perform an experiment. We choose $\zeta_1, ..., \zeta_{m(n)}$ uniformly at random from the set $\{0,1\}^{n \times m(n)}$, and we choose $\lambda_1, ..., \lambda_{m(n)}$ uniformly at random from the set $\{0,1\}^{n \times m(n)}$. We will evaluate B_n in two different manners using the random choices. Define the output of gate Γ_b^a to be ρ_b^a. For each k, we call η_k the input to Γ_k^1.

For the first evaluation of B_n we do the following. For each k we set ρ_k^2 to be ζ_k. Notice that by fixing the $\rho_1^2, ..., \rho_{m(n)}^2$ we have completely determined $\eta_1, .., \eta_{m(n)}$, the inputs to each oracle gate. Further, we have determined the output of the circuit B_n, which we will denote the output by the pair (η_k, μ), where μ is B_n's guess of $s(\eta_k)$, for some k. Now for each k we set ρ_k^1 to be λ_k.

Observe that these choices do not correspond to randomly choosing a function $t \in \mathcal{F}^n$, and then running B_n on the oracle $t \circ t$. In fact, it may be the case that these random choices do not even correspond to functions. For example, it may be the case that $\lambda_j = \lambda_k$, but that $\zeta_j \neq \zeta_k$, and therefore the choices would not even correspond to a function, as functions are single-valued. However, had these choices corresponded to a random function, then the probability that the output of the circuit, (η_k, μ), was correct, is the probability that $\mu = \lambda_k$. We note that the probability that $\mu = \lambda_k$ is exactly $\frac{1}{2^n}$. Let $\mathcal{E}_1$ be the event that $\mu = \lambda_k$, during the first evaluation of B_n.

As stated earlier, it is clearly the case that the above experiment does not correspond to choosing a function $\hat{s}$ uniformly at random from $\mathcal{F}^n$, and then using $\hat{s} \circ \hat{s}$ as an oracle for B_n. We will correct this in the second evaluation of B_n. Further, we will now bound the number of cases in which the first evaluation fails to correspond to choosing a function $\hat{s} \in \mathcal{F}^n$, and then using $\hat{s} \circ \hat{s}$ as an oracle.

We now describe the second evaluation of B_n, and we note that it simulates B_n running on the input $\hat{s} \circ \hat{s}$. We consider the gates of B_n in the order $\Gamma_1^1, \Gamma_1^2, ..., \Gamma_{m(n)}^1, \Gamma_{m(n)}^2$. We need to set the outputs of the gates Γ_b^a in fashion which is equivalent to how they would be set if we were to truly choose an $\hat{s}$ uniformly at random from $\mathcal{F}^n$, and then use $\hat{s} \circ \hat{s}$ as an oracle. To accomplish this we set the outputs of the oracle gates to be the random choices, as before, except when the random choice would not correspond to a function of the form $\hat{s} \circ \hat{s}$. This occurs when the random choice for the output is not consistent with previously assigned random outputs of previous gates. If such an exception arises, then we fix the output of the gate so that it is consistent with the outputs of previous gates. Formally, we set ρ_k^1 to be λ_k, unless there exists a $\rho_\ell^1 = \rho_k^1$, for $\ell < k$, and in this case we set ρ_k^1 to be ρ_ℓ^2. Similarly, we set ρ_k^2 to be ζ_k, unless there exists an $\ell < k$ where $\rho_k^1 = \rho_\ell^1$

or $\rho_k^1 = \eta_\ell$, in which case we set ρ_k^2 to be ρ_ℓ^2 or ρ_ℓ^1 respectively.

When there exists a k such that we do not set ρ_k^1 to be λ_k, or we do not set ρ_k^2 to be ζ_k then we say a *collision* has occurred. Intuitively, this is when a random choice was not consistent with a function of the form $\hat{s} \circ \hat{s}$, and so the output of a gate was set so that it was consistent with previous gates outputs.

Notice that it is only when collisions occur that the first evaluation of B_n differs from the second evaluation of B_n.

Let $\mathcal{E}_2$ be the event of a collision occurring in second evaluation of B_n. We bound the probability of event $\mathcal{E}_2$. Clearly, if all of the η's and λ's are distinct, then no collision can occur. We bound this probability.

$$\begin{aligned}
\Pr_{\substack{\zeta_1,\ldots,\zeta_{m(n)} \in \{0,1\}^{n\cdot m(n)} \\ \lambda_1,\ldots,\lambda_{m(n)} \in \{0,1\}^{n\cdot m(n)}}} [|\{\lambda_j, \eta_k | 1 \le j,k \le m(n)\}| < 2m(n)] &\le \frac{m(n)}{2^n} + \frac{m(n)+1}{2^n} + \cdots + \frac{2m(n)}{2^n} \\
&= \frac{(m(n))^2 + (1 + \cdots + m(n))}{2^n} \\
&= \frac{(m(n))^2 + (m(n)^2 + m(n))}{2^{n+1}} \\
&= \frac{2m(n)^2 + m(n)}{2^{n+1}}
\end{aligned}$$

Define $\mathcal{E}_3$ to be the event that B_n outputs (η_k, μ), and $\mu = \rho_k^1$ during the second evaluation of B_n. Clearly, this is only the case when the output of gate Γ_k^1 is equal to μ. Therefore, $\mathcal{E}_3 \subset \mathcal{E}_1 \cup \mathcal{E}_2$, and this implies that:

$$\begin{aligned}
\Pr \mathcal{E}_3 &\le \Pr \mathcal{E}_1 + \Pr \mathcal{E}_2 \\
&= \frac{1}{2^n} + \frac{2m(n)^2 + m(n)}{2^{n+1}} \\
&= \frac{2 + 2m(n)^2 + m(n)}{2^{n+1}}
\end{aligned}$$

Notice that by the definition of $\mathcal{E}_3$, and the design of the experiment that

$$\Pr_{\hat{s} \in \mathcal{F}^n} [B_n(\hat{s} \circ \hat{s}) = (\eta_k, \hat{s}(\eta_k))] = \Pr[\mathcal{E}_3] \le \frac{2 + 2m(n)^2 + m(n)}{2^{n+1}}.$$

We now remember that we are interested in the case in which $\hat{s}$ is chosen uniformly at random for the set $\widehat{S}^n$, and not the set $\mathcal{F}^n$. We use our bound for the latter case to derive a bound for the former.

$$\begin{aligned}
\Pr_{\hat{s}\in\widehat{S}^n}[\mathcal{E}_3] &= \Pr_{\hat{s}\in\mathcal{F}^n}[\mathcal{E}_3|\hat{s}\in\widehat{S}^n] \\
&\le \frac{\Pr_{\hat{s}\in\mathcal{F}^n}[\mathcal{E}_3]}{\Pr_{\hat{s}\in\widehat{S}^n}} \\
&= \frac{\frac{2+2m(n)^2+m(n)}{2^{n+1}}}{\frac{1}{n^c}} \\
&= \frac{n^c(2+2m(n)^2+m(n))}{2^{n+1}} \\
&\le \frac{n^c\cdot m(n)^3}{2^n}
\end{aligned}$$

$\ldots\square$

We now prove the lemma which corresponds to Lemma 5.8. Remember that this lemma is only used to later prove a lemma which corresponds to Lemma 5.7.

Lemma 6.3 *For every n let $S^n\subset\mathcal{F}^n$ be a set such that $\frac{|S^n|}{|\mathcal{F}^n|}\ge\frac{1}{n^c}$, for some constant $c>0$. Let $\{C_n\}$ be an infinite family of polynomial in n sized decision-circuits. We show that for any $d>0$, there exists an $r>0$ such that for all sufficiently large n:*

$$\Pr_{(f,s_1,\ldots,s_{n^r})\in\mathcal{F}^n\times(S^n)^{n^r}}\left[\left|\frac{1}{n^r}\sum_{i=1}^{n^r}C_n(f\square s_i)-\Pr_{C_n}(\mathcal{F}^n)\right|>\frac{1}{n^d}\right]<\frac{1}{2^{n/3}}.$$

Proof: We will assume that r is fixed, and later we will show how to determine r's value.

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

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

In the second experiment pick random $(f, s_1, ..., s_n^r) \in S^n \times (\mathcal{F}^n)^{n^r}$, and evaluate $C_n(f \Box s_i)$ for each i, where $1 \leq i \leq n^r$. Define the event E_2 to be:

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

Using the Chernoff Bound (Lemma 5.4) we choose an r where the probability of event E_1 occurring in the first experiment is less than $\frac{1}{2^{\frac{n}{2}}}$. 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. The lemma will follow.

We will perform a third experiment, in which we model both of the first two experiments by considering two different methods of evaluating the circuit C_n, n^r times. However, in the third experiment we use a modified form of C_n. To modify C_n, we replace every function gate, g_j, which corresponds to the function f_i or $f \Box s_i$ in experiments one and two respectively, with two function gates, I_j and O_j. In the first experiment the gate I_j would correspond to the identity function and O_j would correspond to the function f_i. In the second experiment both I_j and O_J would correspond to the functions $f \oplus s_i$. We redirect g_j's input to I_j; we take the output of I_j and use it as the input for O_j; and we replace the outputs of g_j with the output of O_j. Notice that the behavior of C_n has not been modified. We have simply modified the representation of the circuit. We let I_j^i and O_j^i correspond to the gates which replaced g_j, in the manner previously mentioned, during the evaluation of $C_n(f_i)$ and $C_n(f \Box s_i)$, respectively in our models of experiments one and two.

First we choose $s_1, ..., s_{n^r} \in S^n$; $\beta_1^1, ..., \beta_{m(n)}^1, ..., \beta_1^{n^r}, ..., \beta_{m(n)}^{n^r} \in \{0,1\}^{n \times m(n) \times n^r}$; and then we choose $\gamma_1^1, ..., \gamma_{m(n)}^1, ..., \gamma_1^{n^r}, ..., \gamma_{m(n)}^{n^r} \in \{0,1\}^{n \times m(n) \times n^r}$. We represent the input to the gate I_b^a as α_b^a; we represent the input to the gate O_b^a as $\widehat{\beta}_b^a = I_b^a(\alpha_b^a)$; and we represent the output of the gate O_b^a as $\widehat{\gamma}_b^a$.

We can now model the first experiment in experiment three. Rather than performing n^r evaluations of C_n using a different random function as the oracle for each evaluation, as in experiment one; we equivalently consider n^r evaluations of C_n, where in each evaluation

i and for each j we assign the random element $\gamma_i^j \in \{0,1\}^n$ to $\widehat{\gamma}_b^a$, the output of O_j^i. Notice that by fixing the outputs of each gate we have completely determined the behaviour of the circuit. Further, notice that because of the way we have set the outputs to the gates O_j^i, we have perfectly modeled experiment one. Let $\mathcal{E}_1$ be the event in experiment three which corresponds to event E_1 in experiment one.

Observe that we cannot model the second experiment in the same manner, for we are dealing with functions of the form $f \square s_i = (f \oplus s_i) \circ (f \oplus s_i)(x)$. An example of the problems which can occur is as follows: it is possible for the gates g_i and g_j to be queried on two different inputs $\alpha_i \neq \alpha_j$, but it may be the case that $(f \oplus s_i)(\alpha_i) = (f \oplus s_i)(\alpha_j)$, and in this case the outputs of the two gates must be identical. However, the chances of randomly assigning the same output to both gates is negligibly small. We will remedy this problem, and present a model of experiment two which is similar to our model of experiment one.

We consider the gates in the following order: $I_1^1, O_1^1, \cdots, I_{m(n)}^1, O_{m(n)}^1, \cdots, I_1^{n^r}, O_1^{n^r}, \cdots, I_{m(n)}^{n^r}, O_{m(n)}^{n^r}$. If the input α_b^a of gate I_b^a is not equal to any α_d^c or $\widehat{\beta}_d^c$ for any $(c,d) < (a,b)$, then we assign its output to be $\widehat{\beta}_b^a \leftarrow \beta_b^a$. Similarly, if the input $\widehat{\beta}_b^a$ of gate O_b^a is not equal to any α_d^c or $\widehat{\beta}_d^c$ for any $(c,d) < (a,b)$, then we assign its output to be $\widehat{\gamma}_b^a \leftarrow \gamma_b^a$.

Alternatively, if the input α_b^a of gate I_b^a is equal to α_d^c or $\widehat{\beta}_d^c$, for some $(c,d) \leq (a,b)$, then we assign its output to be $\widehat{\beta}_b^a \leftarrow f(\alpha_b^a) \oplus s^a(\alpha_b^a)$. Similarly, if the input $\widehat{\beta}_b^a$ of gate O_b^a is equal to α_d^c or $\widehat{\beta}_d^c$, for some $(c,d) \leq (a,b)$, then we assign its output to be $\widehat{\gamma}_b^a \leftarrow f(\widehat{\beta}_b^a) \oplus s^a(\widehat{\beta}_b^a)$. These n^r evaluations of C_n now perfectly model experiment two. Let $\mathcal{E}_2$ be the event in experiment three which corresponds to E_2 in experiment two.

It is now easy to see that our view of assigning outputs to the function gates, as in the model of experiment one, goes wrong only if one of the gates O_b^a is queried on an input which was previously queried for a gate O_d^c or I_d^c, for some $(c,d) \neq (a,b)$. We formalize this by defining a collision. Given our random choices in experiment three, we say that a *collision* has occurred if there exists $(a,b) \neq (c,d)$ s.t. $\widehat{\beta}_b^a = \widehat{\beta}_d^c$ or $\widehat{\beta}_b^a = \alpha_d^c$. Let $\mathcal{E}_3$ be the event that a collision has occurred in our model of experiment two.

Given the new perspective of the experiments, observe that $\mathcal{E}_2 \subset \mathcal{E}_1 \cup \mathcal{E}_3$, and this implies that $\Pr[\mathcal{E}_2] \leq \Pr[\mathcal{E}_1] + \Pr[\mathcal{E}_3]$. We will now show that the $\Pr[\mathcal{E}_3] \leq \frac{1}{2^{n/2}}$, and this combined with the fact that $\Pr[E_1] \leq \frac{1}{2^{n/2}}$, and thus that $\Pr[\mathcal{E}_1] \leq \frac{1}{2^{n/2}}$, implies that $\Pr[\mathcal{E}_2] \leq \frac{1}{2^{n/3}}$ and thus that $\Pr[E_2] \leq \frac{1}{2^{n/3}}$. This proves the lemma.

It remains to show that $\Pr[\mathcal{E}_3] \leq \frac{1}{2^{n/2}}$. In order to do this we consider a forth experiment in which we fix $f \in \mathcal{F}^n$. We then draw a function s uniformly at random from the set S^n. We then run the circuit C_n on the oracle $f \Box s$, by having the gates I_j and O_j compute the function $f \oplus s$, for each j.

Remember that α_1^i is completely determined by C_n, but for $j > 1$ the choice of α_j^i can be dependent on $\widehat{\gamma}_1^i, \widehat{\gamma}_2^i, ..., \widehat{\gamma}_{j-1}^i$. Further, $\alpha_j^i \neq \alpha_k^i$, for $j \neq k$, by the construction of C_n. In experiment four, we will consider the probabilities involved in four cases. We show that in first three cases the probability is trivially bounded, and show that in the last case reduces to the problem taken care of in Lemma 6.2. Before describing the cases we define a set used in the four cases. The set is $\widehat{S}^n = \{f \oplus s_i | s_i \in S^n\}$.

Case 1: There exists an i, j, k, l, where $k = i$, such that $\widehat{\beta}_j^i = \widehat{\beta}_l^k$.

Fix $f \in \mathcal{F}^n$, and consider:

$$\Pr_{s_i \in \mathcal{S}^n}[\widehat{\beta}_j^i = \widehat{\beta}_l^k] \quad = \quad \Pr_{s_i \in \mathcal{S}^n}[f(\alpha_j^i) \oplus s_i(\alpha_j^i) = f(\alpha_l^k) \oplus s_k(\alpha_l^k)]$$

Notice that when $i = k$ that

$$\Pr_{s_i \in \mathcal{S}^n}[f(\alpha_j^i) \oplus s_i(\alpha_j^i) = f(\alpha_l^k) \oplus s_k(\alpha_l^k)] = \Pr_{\hat{s}_i \in \widehat{S}^n}[\hat{s}_i(\alpha_j^i) = \hat{s}_i(\alpha_l^k)].$$

A simple argument shows that this probability is bounded by $\frac{m(n)^2 \cdot n^c}{2^n}$. To notice this, observe that given a function $\hat{s}$, drawn uniformly at random from $\mathcal{F}^n$, and given $m(n)$ queries, that for any two distinct queries a and b, the probability that $\hat{s}(a) = \hat{s}(b)$ is less than $\frac{m(n)^2}{2^n}$. Then notice when $\hat{s}$ is drawn uniformly at random from $\widehat{S}^n$ instead of $\mathcal{F}^n$, we can easily bound the probability by $\frac{m(n)^2 \cdot n^c}{2^n}$.

Case 2: There exists an i, j, k, l, where $k \neq i$, such that $\widehat{\beta}_j^i = \widehat{\beta}_l^k$.

Fix $f \in \mathcal{F}^n$, and consider.

$$\Pr_{s_i \in \mathcal{S}^n}[\widehat{\beta}_j^i = \widehat{\beta}_l^k] \quad = \quad \Pr_{s_i \in \mathcal{S}^n}[f(\alpha_j^i) \oplus s_i(\alpha_j^i) = f(\alpha_l^k) \oplus s_k(\alpha_l^k)]$$

Notice that $f(\alpha_l^k) \oplus s_k(\alpha_l^k)$ and α_l^k are completely independent of the choice of s_i. Therefore, we can bound this probability by bounding the probability

$$\Pr_{s_i \in \mathcal{S}^n}[f(\alpha_j^i) \oplus s_i(\alpha_j^i) = \vartheta],$$

where ϑ is a fixed value which has been chosen, non-uniformly, to maximize the above probability. This gives the following probability,

$$\Pr_{s_i \in \mathcal{S}^n}[f(\alpha_j^i) \oplus s_i(\alpha_j^i) = \vartheta] = \Pr_{\hat{s}_i \in \widehat{S}^n}[\hat{s}_i(\alpha_j^i) = \vartheta].$$

This probability is bound to be less than $\frac{m(n) \cdot n^c}{2^n}$. To notice this, observe that given a function $\hat{s}$, drawn uniformly at random from $\mathcal{F}^n$, and given $m(n)$ queries, that for any query a, the probability that $\hat{s}(a) = \vartheta$ is less than $\frac{m(n)}{2^n}$. Then notice when $\hat{s}$ is drawn uniformly at random from $\widehat{S}^n$ instead of $\mathcal{F}^n$, we can easily bound the probability by $\frac{m(n) \cdot n^c}{2^n}$.

Case 3: There exists an i, j, k, l, where $k \neq i$, such that $\widehat{\beta}_j^i = \alpha_l^k$.

Again we fix $f \in \mathcal{F}^n$, and consider:

$$\begin{aligned}
\Pr_{s_i \in \mathcal{S}^n}[\alpha_l^k = \widehat{\beta}_j^i] &= \Pr_{s_i \in \mathcal{S}^n}[\alpha_l^k = f(\alpha_j^i) \oplus s_i(\alpha_l^i)] \\
&= \Pr_{\hat{s}_i \in \widehat{S}^n}[\alpha_l^k = \hat{s}_i(\alpha_j^i)].
\end{aligned}$$

Similarly to the previous case, notice that α_l^k is completely independent of the choice of $\hat{s}_i$. Therefore, we can bound this probability by bounding the following probability,

$$\Pr_{\hat{s}_i \in \widehat{S}^n}[\vartheta' = \hat{s}_i(\alpha_j^i)],$$

where ϑ' is a fixed value which has been chosen, non-uniformly, to maximize the above probability. This probability is bound to be less than $\frac{m(n) \cdot n^c}{2^n}$, for the same reasons as those mentioned in Case 2.

Case 4: There exists an i, j, k, l, where $k = i$, such that $\widehat{\beta}^i_j = \alpha^k_l$.

Again we fix $f \in \mathcal{F}^n$, and consider:

$$\begin{aligned} \Pr_{s_i \in \mathcal{S}^n}[\alpha^k_l = \widehat{\beta}^i_j] &= \Pr_{s_i \in \mathcal{S}^n}[\alpha^k_l = f(\alpha^i_j) \oplus s_i(\alpha^i_l)] \\ &= \Pr_{\hat{s}_i \in \widehat{S}^n}[\alpha^k_l = \hat{s}_i(\alpha^i_j)]. \end{aligned}$$

Clearly, the probability in last cases is bound by a bound on the following probability. First, let $\{B_n\}$ be any polynomial in n sized circuit family, in which every circuit B_n has $m(n)$ oracle gates. We consider the probability when $\hat{s}$ is chosen uniformly at random from the set $\widehat{S}^n$, and that when given as an input the oracle $\hat{s} \circ \hat{s}$ the circuit B_n outputs the pair (α, μ), where α corresponds to one of the oracle queries made by B_n, and $\mu = \hat{s}(\alpha)$. Fortunately, the above bound is given in Lemma 6.2.

Therefore, for each choice of i, j, k and l there are two possible ways in which a collision can occur, and for each way the probability of it occurring is bounded above by $\frac{n^c \cdot m(n)^3}{2^n}$ from the results of Lemma 6.2. There are a total of $\frac{n^{2r}+m(n)^2}{2}$ choices of i, j, k, l, so:

$$\Pr(\mathcal{E}_3) \leq \frac{n^c \cdot m(n)^3 \left(\frac{n^{2r}+m(n)^2}{2}\right)}{2^n} < \frac{1}{2^{\frac{n}{2}}},$$

and the lemma is proved. ...$\Box$

Corollary 6.2 *For each n, let $S^n \subset \mathcal{F}^n$ by any set such that $\frac{|S^n|}{|\mathcal{F}^n|} \geq \frac{1}{n^d}$, for some constant d, and let $\{C_n\}$ be a polynomial sized family of decision-circuits. Then for every constant c, and for all but $\frac{1}{2^{n/3}}$ of the $w \in \mathcal{F}^n$:*

$$\left| \Pr_{C_n}(w \Box S^n) - \Pr_{C_n}(\mathcal{F}^n) \right| < \frac{1}{n^c},$$

for sufficiently large n.

This corollary corresponds directly to Lemma 5.7, but with the composition operator and the set of permutations S^n replaced by the $\Box$ operator and the set of functions S^n.

6.2 Isolation Lemmas

Now by Corollaries 6.1 and 6.2 we can prove both versions of the Isolation Lemmas in Chapter 5 by replacing the composition operator with the $\Box$ operator, and by replacing the PRPGs with PRFGs[1]. For completeness, we state both versions of Isolation Lemmas and their corresponding composition theorems below.

Lemma 6.4 (Box Isolation Lemma – Weak Version) –
There exist fixed polynomials p_1 and p_2 such that for all $0 \leq \epsilon, \delta \leq 1$; polynomials $c_{\mathsf{G}}, c_{\mathsf{H}}$, and s_{F}; and function generators H and G, where $c_{\mathsf{G}}(n)$ and $c_{\mathsf{H}}(n)$ bound from above the size of the circuits which compute G^n and H^n respectively. Define $\mathsf{F} = \mathsf{G} \Box \mathsf{H}$.

Hypothesis: *If there exists a family of decision-circuits $\{C_n\}$, where for each n the circuit C_n is of size $s_{\mathsf{F}}(n)$, and for some $c > 0$ and infinitely many n:*

$$\left| \Pr_{C_n}(\mathsf{F}^n) - \Pr_{C_n}(\mathcal{F}^n) \right| \geq \epsilon\delta\left(2 - \max\{\epsilon, \delta\}\right) + \frac{1}{n^c}.$$

Conclusion: *Then for each sufficiently large n there exists either a decision-circuit Λ_n of size $p_1(n^c \cdot c_{\mathsf{G}}(n))s_{\mathsf{F}}(n)$ for which:*

$$\left| \Pr_{\Lambda_n}(\mathsf{G}^n) - \Pr_{\Lambda_n}(\mathcal{F}^n) \right| \geq \delta + \frac{1}{n^{24c}};$$

or a decision-circuit Υ_n of size $p_2(n^c \cdot c_{\mathsf{G}}(n))s_{\mathsf{F}}(n)$ for which:

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

or a decision-circuit Ξ_n of size $c_{\mathsf{H}}(n) + s_{\mathsf{F}}(n)$ for which:

$$\left| \Pr_{\Xi_n}(\mathsf{G}^n) - \Pr_{\Xi_n}(\mathcal{F}^n) \right| \geq \delta + \frac{1}{n^{6c}}.$$

Corollary 6.3 (Box Theorem) *Let G be a $1-\epsilon$ secure PRPG. Then for each positive integer c, the generator $\mathsf{F} = \underbrace{(\cdots(\mathsf{G}\Box\mathsf{G})\Box\cdots\Box\mathsf{G})}_{c\ times}$ is $1-\theta$ secure, where $\theta = \epsilon^c(2-\epsilon)^{c-1}$.*

[1] Technically the original Isolation Lemma requires a commutative version of Lemma 5.6 and 5.7, but since the $\Box$ operator is commutative the corresponding lemmas follow immediately.

For the theorem below we need a definition for a semi-secure PRFG which is analogous to the definition of a semi-secure PRPG. We will call a PRFG, H, $1-\epsilon$ *semi-secure* if there exists no family of polynomial-sized decision- circuits $\{C_n\}$ which has an advantage of $\epsilon + \frac{1}{\log^c n}$ in distinguishing H^n from $\mathcal{F}^n$, for some constant c and infinitely many n.

Lemma 6.5 (Box Isolation Lemma – Stronger Version) –

There exist a fixed polynomials p_1 and a fixed poly-logarithmic function p_2 such that for all $0 < \delta < 1$; $\epsilon : \mathbb{N} \to [0,1]$, where for all sufficiently large n, $\epsilon(n) < \delta$; polynomials $c_{\mathsf{G}}, c_{\mathsf{H}}$, and s_{F}; and function generators H and G, where $c_{\mathsf{G}}(n)$ and $c_{\mathsf{H}}(n)$ bound from above the size of the circuits which compute G^n and H^n respectively. Define $\mathsf{F} = \mathsf{G}\Box\mathsf{H}$.

Hypothesis: *If there exists a family of decision-circuits $\{C_n\}$, where for each n the circuit C_n is of size $s_{\mathsf{F}}(n)$, and for some $c > 0$ and all sufficiently large n:*

$$\left|\Pr_{C_n}(\mathsf{F}^n) - \Pr_{C_n}(\mathcal{F}^n)\right| \geq \epsilon(n)\delta\,(2 - \max\{\epsilon, \delta\}) + \frac{1}{\log^c n}.$$

Conclusion: *Then for infinitely many n there exists either a decision-circuit Λ_n of size $p_1(\log^c n \cdot c_{\mathsf{G}}(n))s_{\mathsf{F}}(n)$ for which:*

$$\left|\Pr_{\Lambda_n}(\mathsf{G}^n) - \Pr_{\Lambda_n}(\mathcal{F}^n)\right| \geq \delta + \frac{1}{n};$$

or a decision-circuit Υ_n of size $p_2(\log^c n \cdot c_{\mathsf{G}}(n))s_{\mathsf{F}}(n)$ for which:

$$\left|\Pr_{\Upsilon_n}(\mathsf{H}^n) - \Pr_{\Upsilon_n}(\mathcal{F}^n)\right| \geq \epsilon(n) + \frac{1}{\log^{2c} n};$$

or a decision-circuit Ξ_n of size $c_{\mathsf{H}}(n) + s_{\mathsf{F}}(n)$ for which:

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

Theorem 6.1 (Box Theorem – Stronger Version) –

Let G be a $1-\epsilon$ secure PRPG. Then for each $f \in \mathcal{O}(\log\log n)$ there exits a generator $\mathsf{F} = \underbrace{(\cdots(\mathsf{G}\Box\mathsf{G})\Box\cdots\Box\mathsf{G})}_{f(n)\ times}$ which is $1-\theta(n)$ semi-secure, where $\theta(n) = \epsilon^{f(n)}(2-\epsilon)^{f(n)-1}$.

As in the case of the composition operator acting on permutation generators, it is a natural question to ask if the Isolation Lemma's security parameter is still tight for the function generator that results from the $\Box$ operator acting on partially secure PRFGs. We show in the next section that the Isolation Lemma is still tight.

6.3 The Isolation Lemma is Still Tight

We now show that there exist PRFGs H and G which are respectively $(1-\epsilon)$ and $(1-\delta)$ secure, but when boxed together are exactly $(1-\theta)$ secure, where $\theta = \epsilon\delta(2-\max\{\epsilon, \delta\})$. We point out to the reader that there is much in common between this construction and the construction presented in Chapter 5.

The Construction of G and H

We present the construction of G, and note that the construction of H is similar. To simplify the presentation we assume that δ and ϵ are of the form $\frac{1}{2^c}$ or $1-\frac{1}{2^c}$. Let $\widehat{\mathsf{G}} = \{\widehat{\mathsf{G}}^n : \{0,1\}^{\ell(n)} \times \{0,1\}^n \to \{0,1\}^n | n \in \mathbb{N}\}$ be a PRFG. We construct $\mathsf{G}^n = \{\mathsf{G}^n : \{0,1\}^{\ell(n)+c} \times \{0,1\}^n \to \{0,1\}^n | n \in \mathbb{N}\}$ to be $1-\delta$ secure in two steps. We will describe the construction of G^n, and note that the construction of H^n is similar.

First, for each n we set a fraction δ of the keys of G^n to correspond to the zero function, $f(x) = \bar{0}\ \forall x$, and the remainder to correspond to functions chosen from $\widehat{\mathsf{G}}^n$.This is done in two different fashions dependent on the form of δ as described below:

Case 1 ($\delta = 1-\frac{1}{2^c}$) For each $k \in \{0,1\}^{\ell(n)+c}$ we set the function $g_k^n \in \mathsf{G}^n$ to be the zero function, if the first c bits of k *are not* all 0; otherwise let $\bar{k}$ be the last $\ell(n)$ bits of k, and set $g_k^n = \hat{g}_{\bar{k}}^n$, for $\hat{g}_{\bar{k}}^n \in \widehat{\mathsf{G}}^n$.

Case 2 ($\delta = \frac{1}{2^c}$) For each $k \in \{0,1\}^{\ell(n)+c}$ we set the function $g_k^n \in \mathsf{G}^n$ to be the zero function, if the first c bits of k *are* all 0; otherwise let $\bar{k}$ be the last $\ell(n)$ bits of k, and set $g_k^n = \hat{g}_{\bar{k}}^n$, for $\hat{g}_{\bar{k}}^n \in \widehat{\mathsf{G}}^n$.

The second step in the construction of G^n is to ensure that for each $k \in \{0,1\}^{\ell(n)+c}$ we set the value of $g_k^n(g_k^n(\bar{0}))$ in one of two manners, depending on the form of δ, as described below:

Case 1 ($\delta = 1 - \frac{1}{2^c}$) For all keys k set the first c bits of $g_k^n(g_k^n(\bar{0}))$ to 0; the last $n-c$ bits remain as they were in $g_k^n(\bar{0})$.

Case 2 ($\delta = \frac{1}{2^c}$) For all keys k ensure that not all of the first c bits of $g_k^n(g_k^n(\bar{0}))$ are 1. If they are, then set them to be a member of the set $\{0,1\}^c \setminus \{1^c\}$ chosen uniformly at random[2]; the last $n-c$ bits remain as they were in $g_k^n(g_k^n(\bar{0}))$.

It remains to show that G and H are in fact $1-\delta$ and $1-\epsilon$ secure. We note that the proof is similar to the proof presented in Appendix A, and so we will not present it here.

The Adversary

We now show that there exists an adversary which can $\epsilon\delta(2 - \max\{\epsilon, \delta\})$ distinguish $\mathsf{F} = \mathsf{G}\Box\mathsf{H}$, and thereby show that the isolation lemma result is tight. We assume WLOG that $\delta \geq \epsilon$.

Given a function $w \in \mathcal{F}^n$ the adversary, A, accepts in one of the following two conditions, dependent on the form of δ:

Case 1 ($\delta = 1 - \frac{1}{2^c}$) The adversary A accepts *iff* the first c bits of $w(w(\bar{0}))$ are 0;

Case 2 ($\delta = \frac{1}{2^c}$) The adversary A accepts *iff* the first c bits of $w(w(\bar{0}))$ are not 1.

We first consider the accepting probability of A if w was chosen uniformly from F^n. In this case w is the result of applying the box operator to two functions chosen randomly from each of the respective generators G^n and H^n. We partition the functions of G^n into

[2]In practice a PRFG p would be used to set the first c bits of $g_k^n(g_k^n(\bar{0}))$. This would be done by setting the first c bits of $g_k^n(g_k^n(\bar{0})$ to be equivalent be the first consecutive set of c bits in $p(k)$ which were not all 1. This allows the first c bits to be computed, and discards the need to store them, which would not be possible as it would require an exponential amount of storage.

two sets. We let I_{G^n} represent the set of all zero functions in G^n, and we let P_{G^n} represent the remaining functions in G^n. Similarly, we partition H^n into two sets I_{H^n} and P_{H^n}. We now consider the probability that w is accepted based on it being the result of applying the box operator to two functions from the above mentioned partitions.

Case 1 ($w \in I_{\mathsf{G}^n} \square I_{\mathsf{H}^n}$) The adversary will accept w with probability 1. This case occurs with probability $\delta\epsilon$.

Case 2 ($w \in I_{\mathsf{G}^n} \square P_{\mathsf{H}^n}$) The adversary will accept w with probability $\frac{1-\delta}{1-\epsilon}$, which is the probability that A would accept a random function conditioned on the fact that the function w was chosen from P_{H^n}. This case occurs with probability $\delta(1-\epsilon)$.

Case 3 ($w \in P_{\mathsf{G}^n} \square I_{\mathsf{H}^n}$) The adversary will accept w with probability 1. This case occurs with probability $(1-\delta)\epsilon$.

Case 4 ($w \in P_{\mathsf{G}^n} \square P_{\mathsf{H}^n}$) The adversary will accept w with probability $1-\delta$, which is the probability that the result of a random function evaluated at a random point in its range would meet the acceptance criteria of A. This case occurs with probability $(1-\delta)(1-\epsilon)$.

Clearly a random function w will be accepted by A with probability $1-\delta$. Now following the argument made in section 5.1.1 it is clear that the adversary has distinguishing probability of $\epsilon\delta\left(2-\max\{\epsilon,\delta\}\right)$.

6.4 Scalability Issues

Again, our goal is to show that by scaling this construction large enough we can achieve a 1-secure PRPG. We believe that this will be possible by considering the finer notion of security which is proposed in section 5.4. However, unless we can eventually achieve complete security by a construction $\underbrace{(\cdots(\mathsf{G}\square\mathsf{G})\square\cdots\square\mathsf{G})}_{\mathcal{O}(\log n)}$ then this construction will not be useful. This is because the size of the construction becomes larger than any fixed polynomial for sufficiently large n, and thus the generator is not computable in polynomial time. This is due to the fact that the size of the implementation of the construction is exponential in the number of $\Box$ operators which are used in the construction. This becomes obvious by rewriting the construction and substituting the $\Box$ operator with its definition. This may limit the usefulness of this construction in practice, for it may be too slow to practically consider. Further, it may prevent polynomial size constructions of this form from becoming PRFGs.

Chapter 7

The $\diamond$ Operator Generator

In the previous two chapters we have shown that the composition and $\square$ operators are security increasing for PRPGs and PRFGs respectively. Unfortunately, in both cases we have not been able to show that constructions based on them can give 1-secure generators. In this chapter we will show that the $\diamond$ operator generator is security increasing when applied to PRFGs. Further, we will show that it can be used to construct completely secure PRFGs from $1-\delta$ secure PRFGs. We now remind the reader of the definition of the $\diamond$ operator generator.

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

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

We will now show that this operator-generator is security increasing, by modifying the arguments which were presented in Chapters 5 and 6. We will see that the fact that $\diamond$ is an operator-generator, as opposed to an operator, does not have anything more than a cosmetic effect on the proof.

Lemma 7.1 (Diamond Isolation Lemma) –

There exists a fixed polynomial p_2 such that for all $\epsilon, \delta : \mathbb{Z} \to [0,1]$; polynomials $c_{\mathsf{G}}, c_{\mathsf{H}}$, and s_{F}; and function generators H and G, where $c_{\mathsf{G}}(n)$ and $c_{\mathsf{H}}(n)$ bound from above the size of the circuits which compute G^n and H^n respectively. Define $\mathsf{F} = \mathsf{G}\diamond\mathsf{H}$.

Hypothesis: *If there exists a family of decision-circuits $\{C_n\}$, where for each n the circuit C_n is of size $s_{\mathsf{F}}(n)$, and for some $c > 0$ and infinitely many n:*

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

Conclusion: *Then for infinitely many n there exists either a decision-circuit Υ_n of size $p_2(n^c \cdot c_{\mathsf{G}}(n))s_{\mathsf{F}}(n)$ for which:*

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

or a decision-circuit Ξ_n of size $c_{\mathsf{H}}(n) + s_{\mathsf{F}}(n)$ for which:

$$\left|\Pr_{\Xi_n}(\mathsf{G}^n) - \Pr_{\Xi_n}(\mathcal{F}^n)\right| \geq \delta(n) + \frac{1}{n^{6c}}.$$

Notice that in this version of the Isolation Lemma that there is only one circuit which distinguishes G from random, and the resulting circuit is much smaller than the circuit needed to break H. This allows us to iteratively apply the Isolation Lemma to the construction $\underbrace{\mathsf{G}\diamond\cdots\diamond\mathsf{G}}_{p(n)}$ for a $p \in \Omega(\log^2 n) \cap (\cup_{i=1}^{\infty}\mathcal{O}(n^i))$ and achieve complete security.

Theorem 7.1 (Diamond Composition Theorem) –
Let G be a $1-\epsilon$ secure PRFG. Then for each $p \in \Omega(\log^2 n) \cap (\cup_{i=1}^{\infty}\mathcal{O}(n^i))$ the generator $\mathsf{F} = \underbrace{\mathsf{G}\diamond\cdots\diamond\mathsf{G}}_{p(n)}$ is a secure PRFG.

Proof: Similar to the proof of corollary 5.1. ...□

We will now present a proof of Lemma 7.1. As was the case in Chapter 6, the important changes to the proof rely on two technical lemmas. We present these lemmas first, and then give the complete proof. It will become evident that the fact that we are using an operator generator instead of an operator has no significant effect on the proof.

7.1 Two Technical Lemmas

As mentioned in Chapter 6, the proof of security presented in Chapter 5 only makes specific use of the properties of composition and permutations in Lemmas 5.6 and 5.7. Therefore, we will need to prove corresponding lemmas which correspond to the $\diamondsuit$ operator. In this chapter we shall assume the same standardized circuits and circuit families which were presented in Section 6.1. Specifically, all circuits will never repeat oracle queries, and all circuits C_n in a circuit family $\{C_n\}$ will perform exactly $m(n)$ queries, for some polynomial m.

Below we present the first technical lemma. It corresponds to Lemma 5.6.

Lemma 7.2 *Given any decision-circuit C, for each $f \in \mathcal{F}^n$ and for each $r_1, r_2 \in \{0,1\}^n$:*

$$\Pr_C(f \diamondsuit^n_{r_1 \bullet r_2} \mathcal{F}^n) = \Pr_C(\mathcal{F}^n).$$

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

Corollary 7.1 *Given any decision-circuit C, for each $f \in \mathcal{F}^n$:*

$$\Pr_C(f \diamondsuit \mathcal{F}^n) = \Pr_C(\mathcal{F}^n).$$

We now present the second technical lemma, and from it we will derive a simple corollary which corresponds to Lemma 5.7. Notice that the proof of this lemma is very similar to the proof of Lemma 6.3.

Lemma 7.3 *For every n let $S^n \subset \mathcal{F}^n$ be a non-empty set. Let $\{C_n\}$ be a family of polynomial in n sized decision-circuits. We show that for any $d > 0$, there exists an $r > 0$ such that for all sufficiently large n:*

$$\Pr_{(f,k_1^1,k_1^2,\ldots,k_{n^r}^1,k_{n^r}^2,s_1,\ldots,s_{n^r}) \in \mathcal{F}^n \times \{0,1\}^{(2n)(n^r)} \times (S^n)^{n^r}} \left[\left| \frac{1}{n^r} \sum_{i=1}^{n^1} C_n(f \diamondsuit^n_{k_i^1 \bullet k_i^2} s_i) - \Pr_{C_n}(\mathcal{F}^n) \right| > \frac{1}{n^d} \right] < \frac{1}{2^{n/3}}.$$

Notice that in this lemma there is no restriction on the size of S^n, other than it being non-empty. It is this lack of restriction on the size of S^n which ends up allowing us to iteratively apply the $\diamond$ operator a polynomial number of times, as is stated in the Diamond Composition Theorem (Theorem 7.1). Notice than in similar Lemmas presented in Chapters 5 and 6 (Lemmas 5.8 and 6.3), there is a requirement that $\frac{|S^n|}{|\mathcal{F}^n|} \geq \frac{1}{n^c}$, for some constant c. Therefore, this lemma is a much stronger result.

Proof: We will assume that r is fixed, and later we will show how to determine r's value. First we define two experiments. In the first experiment pick random $(f_1, f_2, ..., f_n^r) \in (\mathcal{F}^n)^{n^r}$, and evaluate $C_n(f_i)$ for each $i \in \{1, ..., n^r\}$. Define the event E_1 to be:

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

In the second experiment pick random $(f, k_1^1, k_1^2, ..., k_{n^r}^1, k_{n^r}^2, s_1, ..., s_{n^r}) \in \mathcal{F}^n \times \{0,1\}^{(2n)(n^r)} \times (S^n)^{n^r}$, and evaluate $C_n(f \diamond_{k_i^1 \bullet k_i^2} s_i)$ for each i, where $1 \leq i \leq n^r$. Define the event E_2 to be:

$$\left| \frac{1}{n^r} \sum_{i=1}^{n^r} C_n(f \diamond^n_{k_i^1 \bullet k_i^2} s_i) - \Pr_{C_n}(\mathcal{F}^n) \right| > \frac{1}{n^d}.$$

Using the Chernoff Bound (Lemma 5.4) we choose an r where the probability of event E_1 occurring in the first experiment is less than $\frac{1}{2^{\frac{n}{2}}}$. 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. The lemma will follow.

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

We choose $s_1, ..., s_r \in S^n$; $\gamma_1^1, ..., \gamma_{m(n)}^1, ..., \gamma_1^{n^\gamma}, ..., \gamma_{m(n)}^{n^r} \in \{0,1\}^{n \cdot m(n) \cdot n^r}$; and $k_1^1, k_1^2, ..., k_{n^r}^1, k_{n^r}^2 \in \{0,1\}^{2n \cdot n^r}$. Let g_j^i represent the j^{th} oracle-gate of C_n when evaluating f_i in experiment one, or $f \diamond^n_{k_i^1 \bullet k_i^2} s_i$ 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 $g_1^1, g_2^1, ..., g_{m(n)}^1, ..., g_1^{n^r}, ...g_{m(n)}^{n^r}$.

We model the first experiment of performing n^r evaluations of C_n, where in each evaluation we set C_n's oracle to be a random function by equivalently considering n^r evaluations of C_n, where in the evaluation of f_j we independently assign the element γ_i^j to O_j^i, the output of each oracle gate g_j in C_n. Notice that by fixing, for each i and each j, the value of O_j^i, we have completely determined the behaviour of the circuit C_n in each of the evaluations. Further, notice that because of the way we have set the outputs of the gates, we have perfectly modeled experiment one. Let $\mathcal{E}_1$ be the event in experiment three which corresponds to event E_1 in experiment one.

Observe that we could model the second experiment in the same manner if we could guarantee that f would never be queried on the same input. Since our circuits never repeat queries, there is no worry that C_n will every query $f\diamond^n_{k_i^1\bullet k_i^2} s_i$ on the same input. Unfortunately, in our experiment there is the possibility that for $i \neq j$ that C_n will perform a query α on input $f\diamond^n_{k_i^1\bullet k_i^2} s_i$ and a query β on input $f\diamond^n_{k_j^1\bullet k_j^2} s_j$ where $\alpha \oplus k_i^1 = \beta \oplus k_j^1$, and the result will be that f is queried on the same input twice. However, we can model experiment two in a method similar to which we modeled experiment one.

For each evaluation of $C_n(f\diamond^n_{k_i^1\bullet k_i^2} s_i)$ we set the output O_ℓ^i to be γ_ℓ^i, for each ℓ such that $1 \leq \ell \leq m(n)$. 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. Now that the inputs and outputs have been fixed, the only way the f can be queried on the same input twice is if we have bad choices for our k_i^1's. Now for each i, we check if there exists a pair (a, b) and a j, where $(a, b) < (i, j)$, such that $I_j^i \oplus k_i^1 = I_b^a \oplus k_a^1$; 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^1, and therefore f is inadvertently being queried twice on the same input, and we therefore have to make sure that the two query responses are consistent. If a collision has occurred, then we reevaluate $C_n(f\diamond^n_{k_i^1\bullet k_i^2} s_i)$ in the following manner. We reevaluate the gates $g_1^i, ..., g_{m(n)}^i$ in that order. For each gate g_j^i we consider its input I_j^i, and the input to f which is $I_j^i \oplus k_i^1$. If there exists a pair $(a, b) < (i, j)$ such that $I_b^a \oplus k_a^1 = I_j^i \oplus k_i^1$, then f is being queried on an input which has previously been performed in the experiment. Therefore, we set the output O^i_j to be $f(I^i_j \oplus k^1_i) \oplus s_i(I^i_j \oplus k^2_i)$, and force the oracle-gates to respond consistently to queries on f. If no such pair (a, b) existed then the random response was a consistent response, and therefore we set $O^i_j \leftarrow \gamma^i_j$. Notice that this method of evaluating C_n, perfectly models experiment two. Let $\mathcal{E}_2$ be the event in experiment three which 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 $\mathcal{E}_2 \subset \mathcal{E}_1 \cup \mathcal{E}_3$, which implies that $\Pr(\mathcal{E}_2) \leq \Pr(\mathcal{E}_1) + \Pr(\mathcal{E}_2)$. Since the probability of E_1 is less than $\frac{1}{2^{n/2}}$, in experiment one, and thus the probability of $\mathcal{E}_1$ is less than $\frac{1}{2^{n/2}}$, in experiment three; it suffices to show that the probability of event $\mathcal{E}_3$ in experiment three is less than $\frac{1}{2^{n/2}}$ in order to prove that the probability of $\mathcal{E}_2$, in experiment three, is less than $\frac{1}{2^{n/3}}$, which implies that the probability of E_2, in experiment two, is less than $\frac{1}{2^{n/3}}$. This proves the lemma.

We now bound from above the probability of event $\mathcal{E}_3$. Let

$$L_i = \left\{ \left((I^i_j \oplus k^1_i), (\gamma^i_j \oplus s_i(I^i_j \oplus k^2_i)) \right) | 1 \leq j \leq m(n) \right\}.$$

The set L_i represents the pairs $(x, f(x))$ which would be needed in order to calculate the function $f \diamond^n_{k^1_i \bullet k^2_i} s_i$ for the queries which are made by C_n during the evaluation of $C_n(f \diamond^n_{k^1_i \bullet k^2_i} s_i)$. Notice that $(I^i_j \oplus k^1_i)$ corresponds to the value at which f is evaluated, given that the query I^i_j is made to the function $f \diamond^n_{k^1_i \bullet k^2_i} s_i$. Similarly, $(\gamma^i_j \oplus s_i(I^i_j \oplus k^2_i))$ corresponds to the value of f evaluated on $(I^i_j \oplus k^1_i)$ given that $(f \diamond^n_{k^1_i \bullet k^2_i} s_i)(I^i_j \oplus k^1_i) = \gamma^i_j$. Since all of C_n's queries are unique it is the case that a repeated query to f cannot be strictly contained in the evaluation of $C_n(f \diamond^n s_i)$, and therefore $|L_i| = m(n)$ for each i. We now consider the probability that during the evaluation of $C_n(f \diamond^n_{k^1_i \bullet k^2_i} s_i)$ that a collision occurs. We define the set

$$T_i = \left\{ I^i_j \oplus x | 1 \leq j \leq m(n) \wedge (x, f(x)) \in \cup_{\ell=1}^{i-1} L_\ell \right\},$$

which contains all of the possible values of k^1_i which will cause a collision. We observe that $|T_i| \leq m(n)^2 \cdot i$. Therefore the probability of having a collision during the evaluation of $C_n(f \Diamond^n_{k_i^1 \bullet k_i^2} s_i)$ is at most $\frac{m(n)^2 \cdot i}{2^n}$. 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 evaluations of $C_n(f \Diamond^n_{k_i^1 \bullet k_i^2} s_i)$ for each i, where $1 \leq i \leq n^r$. Therefore:

$$\begin{aligned}
\Pr[E_3] &\leq \frac{m(n)^2}{2^n}(1 + 2 + \cdots + n^r) \\
&= \left(\frac{m(n)^2}{2^n}\right)\left(\frac{n^r(n^r+1)}{2}\right) \\
&= \frac{m(n)^2 n^{2r} + n^r m(n)^2}{2^{n+1}} \\
&\leq \frac{1}{2^n}
\end{aligned}$$

This proves the lemma. ...□

Corollary 7.2 *For each n, let $S^n \subset \mathcal{F}^n$ be a non empty set, and let $\{C_n\}$ be a polynomial sized family of decision-circuits. Then for every constant c, and for all but $\frac{1}{2^{n/3}}$ of the $w \in \mathcal{F}^n$:*

$$\left| \Pr_{C_n}(w \Diamond S^n) - \Pr_{C_n}(\mathcal{F}^n) \right| < \frac{1}{n^c},$$

for sufficiently large n.

This corollary corresponds to Lemma 5.7, but is actually much stronger. Since there are no restrictions on the size of S_n we are able to prove a lemma corresponding to Lemma 5.9, as a direct consequence of Corollary 7.2

7.2 Proof Of Lemma 7.1

Assume that there exists a polynomial-sized decision-circuit family $\{C_n\}$ which for some constant $c > 0$ and infinitely many n distinguishes F^n from $\mathcal{F}^n$ with probability at least $\epsilon(n)\delta(n) + \frac{1}{n^c}$.

Lemma 7.4 *For $i > 0$ and for each n let*

$$K_n(i) = \Pr_{C_n}(\mathcal{F}^n) + \frac{1}{n^i} \quad \text{and} \quad L_n(i) = \Pr_{C_n}(\mathcal{F}^n) - \frac{1}{n^i}.$$

Let

$$S^n(i) = \left\{ w \in \mathcal{F}^n \,\middle|\, \Pr_{C_n}(\mathsf{G}^n \Diamond w) \geq K_n(i) \right\} \quad \text{and} \quad T^n(i) = \left\{ w \in \mathcal{F}^n \,\middle|\, \Pr_{C_n}(\mathsf{G}^n \Diamond w) \leq L_n(i) \right\}.$$

Then given i,j:

$$\Pr_{w \in \mathcal{F}^n}(w \in S^n(i)) \leq \frac{1}{n^j} \quad \text{and} \quad \Pr_{w \in \mathcal{F}^n}(w \in T^n(i)) \leq \frac{1}{n^j},$$

for sufficiently large n.

Proof: This lemma follows directly from Corollary 7.2 as $\mathsf{G}^n \subset \mathcal{F}^n$. ...□

Notice the similarities between this lemma, and Lemma 5.9. They are similar, but in Lemma 5.9 there exists the possibility of a family of circuits $\{\Lambda_n\}$, which does not exist in this lemma. It is the lack of this circuit family which permits the large number of applications of the Diamond Operator Generator in the Diamond Composition Theorem, and thereby allows us to attain a PRFG from a partially secure PRFG. Notice that the lack of the family of circuits $\{\Lambda\}$ in this lemma, as opposed to Lemma 5.9, is made possible due to the lack of restrictions on the size of S^n in Corollary 7.2, which is a direct consequence of the lack of restrictions on the size of S^n in Lemma 7.3.

Lemma 7.5 *Either there exists a family of decision-circuits $\{\Xi_n\}$, where for each n the circuit Ξ_n is of size $c_{\mathsf{H}}(n) + s_{\mathsf{F}}(n)$; and for infinitely many n:*

$$\left| \Pr_{\Xi_n}(\mathsf{G}^n) - \Pr_{\Xi_n}(\mathcal{F}^n) \right| \geq \delta(n) + \frac{1}{n^{6c}};$$

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

$$\left| \Pr_{C_n}(\mathsf{G} \Diamond h^n) - \Pr_{C_n}(\mathcal{F}^n) \right| < \delta(n) + \frac{1}{n^{6c}}.$$

Proof: If a family of circuits $\{\Xi_n\}$ with the above properties exists, then we are done. Otherwise, there exists no such family of circuits. Assume that for the circuit family $\{C_n\}$, there exists a set $\{h^n \in \mathsf{H}^n | n \in \mathbb{N}\}$ such that, for all sufficiently large n it is the case that $|\Pr_{C_n}(\mathsf{G}^n \diamond h^n) - \Pr_{C_n}(\mathcal{F}^n)| \geq \delta(n) + \frac{1}{n^{6c}}$. We create the decision-circuit family $\{D_n\}$, where $D_n(w) = C_n(w \diamond h^n)$.

$$\begin{aligned}
\Pr_{D_n}(\mathsf{G}^n) - \Pr_{D_n}(\mathcal{F}^n) &= \Pr_{C_n}(\mathsf{G}^n \diamond h^n) - \Pr_{C_n}(\mathcal{F}^n \diamond h^n) \\
&= \Pr_{C_n}(\mathsf{G}^n \diamond h^n) - \Pr_{C_n}(\mathcal{F}^n) \qquad \text{(Lemma 7.1)} \\
&\geq \delta(n) + \frac{1}{n^{6c}}
\end{aligned}$$

Letting $D_n = \Xi_n$ we have a constructed a family of decision-circuits $\{\Xi_n\}$. This contradicts the fact that there is no decision-circuit family of size $c_{\mathsf{H}}(n) + s_{\mathsf{F}}(n)$, such that for infinitely many n:

$$\left| \Pr_{\Xi_n}(\mathsf{G}^n) - \Pr_{\Xi_n}(\mathcal{F}^n) \right| \geq \delta(n) + \frac{1}{n^{6c}},$$

proving the lemma. ...□

7.2.1 Main Argument

We note that the Main Argument for the Diamond Isolation Lemma is similar, but easier than the Main Argument for the Composition Isolation Lemma presented earlier in section 5.3.1. One major difference is that it performs more sampling, and therefore achieves bounds under the normal security definition of a partially secure PRFG, and not under the semi-secure definition of PRFGS. Another difference is that there is no need to assume the existence of a family of circuits $\{\Lambda_n\}$, due to Lemma 7.4. The final difference is that we present the construction of only one type of circuit in this Main Argument. Unlike the Main Argument in Chapter 5, there is no need to present a circuit which flips biased coins. This is because for a random $w \in \mathcal{F}^n$, Lemma 7.4 constrains the value of $\Pr_{C_n}[\mathsf{G}^n \diamond w]$ to be much closer to $\Pr_{C_n}[\mathcal{F}^n]$, then is the case for the constraint imposed on the distance between $\Pr_{C_n}[\mathsf{G}^n \circ w]$ and $\Pr_{C_n}[\mathcal{F}^n]$, by Lemma 5.9.

Let $\bar{\epsilon}_n = \epsilon(n) + \frac{1}{n^c}$ and similarly let $\bar{\delta}_n = \delta(n) + \frac{1}{n^c}$. Then the family of circuits, $\{C_n\}$, is capable of distinguishing $\mathsf{G}^n \diamond \mathsf{H}^n$ from $\mathcal{F}^n$ with probability at least $\bar{\epsilon}_n \bar{\delta}_n$, for infinitely many n, as this value is strictly smaller than $\epsilon(n)\delta(n) + \frac{1}{n^c}$, for sufficiently large n. We assume that there exists no family of circuits $\{\Xi_n\}$, where each circuit Ξ_n is of size $c_{\mathsf{H}}(n) + s_{\mathsf{F}}(n)$, such that for infinitely many n:

$$\left| \Pr_{\Xi_n}(\mathsf{H}^n) - \Pr_{\Xi_n}(\mathcal{F}^n) \right| \geq \delta(n) + \frac{1}{n^{6c}}.$$

From the above assumption and Lemma 7.5, we know that for all sufficiently large n and all $h^n \in \mathsf{H}^n$:

$$\left| \Pr_{C_n}(\mathsf{G} \diamond h^n) - \Pr_{C_n}(\mathcal{F}^n) \right| < \delta(n) + \frac{1}{n^{6c}}. \tag{7.1}$$

From Lemma 7.4 we know that for

$$K_n = \Pr_{C_n}(\mathcal{F}^n) + \frac{1}{n^{5c}} \quad \text{and} \quad S^n = \left\{ w \in \mathcal{F}^n \,\middle|\, \Pr_{C_n}(\mathsf{G}^n \diamond w) \geq K_n \right\}, \tag{7.2}$$

that $\Pr_{w \in \mathcal{F}^n}(w \in S^n) \leq \frac{1}{n^{7c}}$, for all sufficiently large n. Similarly we know that for

$$L_n = \Pr_{C_n}(\mathcal{F}^n) - \frac{1}{n^{5c}} \quad \text{and} \quad T^n = \left\{ w \in \mathcal{F}^n \,\middle|\, \Pr_{C_n}(\mathsf{G}^n \diamond w) \leq L_n \right\}, \tag{7.3}$$

that $\Pr_{w \in \mathcal{F}^n}(w \in T^n) \leq \frac{1}{n^{7c}}$, for all sufficiently large n.

We will construct a family of decision-circuits $\{\Upsilon_n\}$, where each circuit Υ_n is of size $p_2(n^c \cdot c_{\mathsf{G}}(n)) s_{\mathsf{F}}(n)$, such that for infinitely many n:

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

proving the lemma. In the sequel we show how to construct Υ_n for each n such that both $|\Pr_{C_n}(\mathsf{F}^n) - \Pr_{C_n}(\mathcal{F}^n)| \geq \epsilon(n)\delta(n) + \frac{1}{n^c}.$, and n is sufficiently large for all inequalities in the following proof to hold true.

We first give an overview of the proof. We will construct a circuit B_n (via an intermediate circuit A_n) which, given as input a function w, almost surely approximates the value $\Pr_{C_n}(\mathsf{G}^n \diamond w)$. From Equations 7.2 and 7.3, we know that for almost all random functions $w \in \mathcal{F}^n$ that $\Pr_{C_n}(\mathsf{G}^n \diamond w)$ will be bounded from above by the value K_n, and below by the value L_n.

We show, by means of a simple averaging argument, that it must be the case that for a large fraction of the $h \in \mathsf{H}_n$ (a fraction greater than $\epsilon(n) + \frac{1}{n^{2c}}$) it is the case that $\Pr_{C_n}(\mathsf{G}^n \diamond h)$ falls outside of the range $[L_n, K_n]$. We can then distinguish between H^n and $\mathcal{F}^n$ by computing $B(w)$, and accepting if $B(w)$ is greater than K_n or less than L_n. Because the sets S^n and T^n contain all of the functions w in $\mathcal{F}^n$, for which $\Pr_{C_n}(\mathsf{G}^n \diamond w)$ is less than or greater than L_n and K_n respectively, and since $\frac{|S^n|}{|\mathcal{F}^n|} < \frac{1}{n^{7c}}$ and $\frac{|T^n|}{|\mathcal{F}^n|} < \frac{1}{n^{7c}}$; we know that about a fraction $\frac{2}{n^{7c}}$ of the w will be accepted by $B(n)$.

We construct a probabilistic circuit A_n such that

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

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 $\alpha > 1$ so that $n^\alpha > \kappa(n)$. Using the Chernoff Bound, b is chosen large enough such that:

$$\Pr_{w \in \mathcal{F}^n}\left[\left|A_n(w) - \Pr_{C_n}(\mathsf{G} \diamond w)\right| \geq \frac{1}{n^{7c}}\right] \leq \frac{1}{2^{n^{2\alpha}}},$$

and

$$\Pr_{h^n \in \mathsf{H}^n}\left[\left|A_n(h^n) - \Pr_{C_n}(\mathsf{G} \diamond h^n)\right| \geq \frac{1}{n^{7c}}\right] \leq \frac{1}{2^{n^{2\alpha}}}.$$

By Lemma 5.5 we derandomize A_n to get the circuit B_n for all but $\frac{1}{2^{n^\alpha}}$ of the $w \in \mathcal{F}^n$:

$$\left|B_n(w) - \Pr_{C_n}(\mathsf{G}^n \diamond w)\right| < \frac{1}{n^{7c}}, \tag{7.4}$$

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

$$\left|B_n(h^n) - \Pr_{C_n}(\mathsf{G}^n \diamond h_k^n)\right| < \frac{1}{n^{7c}}, \tag{7.5}$$

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

Let $K'_n = \Pr_{C_n}(\mathcal{F}^n) + \frac{1}{n^{7c}}$, and let $L'_n = \Pr_{C_n}(\mathcal{F}^n) - \frac{1}{n^{7c}}$. We now show that

$$\Pr_{h^n \in \mathsf{H}^n}[B_n(h^n) \geq K'_n \vee B_n(h^n) \leq L'_n] \geq \epsilon(n) + \frac{1}{n^{2c}}.$$

The argument is an averaging argument. We first give an intuitive version of the argument, and then we give all of the details.

We now present the intuitive version of the argument. In this argument we essentially assume that any two values which differ by a value of at most $\frac{1}{\mathrm{poly}(n)}$, for some polynomial poly, are equivalent.

We remember that $B_n(h^n)$ is an approximation of $\Pr_{C_n}(\mathsf{G}^n \diamond h^n)$, and intuitively the two values can be treated as the same. We know that $|\Pr_{C_n}(\mathsf{G}^n \diamond \mathsf{H}^n) - \Pr_{C_n}(\mathcal{F}^n)| > \epsilon(n)\delta(n)$. We partition the functions in H^n into two groups. This first group contains those $h^n \in \mathsf{H}^n$ for which $\Pr_{C_n}(\mathsf{G}^n \diamond h^n) \approx \Pr_{C_n}(\mathcal{F}^n)$. The second group contains the rest of $h^n \in \mathsf{H}^n$ not in the first group. We then observe, by equation 7.1, that for each $h^n \in \mathsf{H}^n$, the value $|\Pr_{C_n}(\mathsf{G} \diamond h^n) - \Pr_{C_n}(\mathcal{F}^n)|$ is less than $\delta(n)$. Using this fact we point out that for a random $h^n \in \mathsf{H}^n$ it must be the case that h^n is in the second group with a probability of at least $\epsilon(n)$, or otherwise $|\Pr_{C_n}(\mathsf{G}^n \diamond \mathsf{H}^n) - \Pr_{C_n}(\mathcal{F}^n)| < \epsilon(n)\delta(n)$.

We know present the exact argument. Assume for contradiction that $\Pr_{h^n \in \mathsf{H}^n}[B_n(h^n) \geq K'_n \vee B_n(h^n) \leq L'_n] < \epsilon(n) + \frac{1}{n^{2c}}$. Let $\mathcal{K}^n \subset \mathsf{H}^n$ be the set of functions $h^n \in \mathsf{H}^n$, for which $B_n(h^n) \geq K'_n$ or $B_n(h^n) \leq L'_n$, and let $\bar{\mathcal{K}}^n$ be its complement.

We now get a contradiction by a simple averaging argument.

$$\begin{aligned}
&\left|\Pr_{C_n}[\mathsf{G} \diamond \mathsf{H}] - \Pr_{C_n}[\mathcal{F}^n]\right| \\
\leq\ & \sum_{h^n \in \mathsf{H}^n} \left(\left|\Pr_{C_n}[\mathsf{G} \diamond h^n] - \Pr_{C_n}[\mathcal{F}^n]\right| \Pr_{f \in \mathsf{H}^n}[f = h^n] \right) \\
=\ & \sum_{h^n \in \mathcal{K}^n} \left(\left|\Pr_{C_n}[\mathsf{G} \diamond h^n] - \Pr_{C_n}[\mathcal{F}^n]\right| \Pr_{f \in \mathsf{H}^n}[f = h^n] \right) \\
&\qquad + \sum_{h^n \in \bar{\mathcal{K}}^n} \left(\left|\Pr_{C_n}[\mathsf{G} \diamond h^n] - \Pr_{C_n}[\mathcal{F}^n]\right| \Pr_{f \in \mathsf{H}^n}[f = h^n] \right) \\
\leq\ & \sum_{h^n \in \mathcal{K}^n} \left(\left|\Pr_{C_n}[\mathsf{G} \diamond h^n] - \Pr_{C_n}[\mathcal{F}^n]\right| \Pr_{f \in \mathsf{H}^n}[f = h^n] \right)
\end{aligned}$$

$$
\begin{aligned}
& + \sum_{h^n \in \bar{\mathcal{K}}^n} \left(\left(\left| B_n(h^n) - \Pr_{C_n}[\mathcal{F}^n] \right| + \frac{1}{n^{7c}} \right) \Pr_{f \in \mathsf{H}^n}[f = h^n] \right) \\
\leq \; & \sum_{h^n \in \mathcal{K}^n} \left(\left| \Pr_{C_n}[\mathsf{G} \diamond h^n] - \Pr_{C_n}[\mathcal{F}^n] \right| \Pr_{f \in \mathsf{H}^n}[f = h^n] \right) + \left(1 - \epsilon(n) - \frac{1}{n^{2c}} \right) \frac{1}{n^{4c}} && (7.6) \\
\leq \; & \sum_{h^n \in \mathcal{K}^n} \left(\left(\delta(n) + \frac{1}{n^{6c}} \right) \left(\Pr_{f \in \mathsf{H}^n}[f = h^n] \right) \right) + \frac{1}{n^{4c}} && (7.7) \\
\leq \; & \left(\epsilon(n) + \frac{1}{n^{2c}} \right) \left(\delta(n) + \frac{1}{n^{6c}} \right) + \frac{1}{n^{4c}} \\
< \; & \epsilon(n)\delta(n) + \frac{1}{n^c}. && (7.8)
\end{aligned}
$$

The inequality in line 7.6 follows from two facts. First that, by assumption, the probability that a random $h^n \in \mathsf{H}^n$ is in $\bar{\mathcal{K}}^n$ is $1 - \epsilon(n) - \frac{1}{n^{2c}}$. Second, that for each h^n in $\bar{\mathcal{K}}^n$, it is the case that $K_n > B_n(h^n) > L_n$, and this implies that $|B_n(h^n) - \Pr_{C_n}(\mathcal{F}^n)| < \frac{1}{n^{5c}}$ by the definitions of K_n and L_n.

The inequality in line 7.7 follows from two facts. First, by assumption, the probability that a random $h^n \in \mathsf{H}^n$ is in $\mathcal{K}^n$ is $(\epsilon(n) + \frac{1}{n^{2c}}$. Second, by equation 7.1, it is the case that for each $h^n \in \mathsf{H}^n$ that $|\Pr_{C_n}(\mathsf{G} \diamond h^n) - \Pr_{C_n}(\mathcal{F}^n)| < \delta(n) + \frac{1}{n^{6c}}$.

The inequality in line 7.8 contradicts the fact that $|\Pr_{C_n}[\mathsf{G} \diamond \mathsf{H}] - \Pr_{C_n}[\mathcal{F}^n]| \geq \epsilon(n)\delta(n) + \frac{1}{n^c}$, and therefore it must be the case that $\Pr_{h^n \in \mathsf{H}^n}[B_n(h^n) \geq K'_n \vee B_n(h^n) \leq L'_n] \geq \epsilon(n) + \frac{1}{n^{2c}}$.

We create the decision circuit $\bar{B}_n(w)$ which accepts w iff $B_n(w) \geq K'_n$ or if $B_n(w) \leq L'_n$.

$$
\begin{aligned}
\left| \Pr_{\bar{B}_n}(\mathsf{H}^n) - \Pr_{\bar{B}_n}(\mathcal{F}^n) \right| \; & \geq \; \epsilon(n) + \frac{1}{n^{2c}} - \Pr_{\bar{B}_n}(\mathcal{F}^n) \\
& \geq \; \epsilon(n) + \frac{1}{n^{2c}} - \frac{2}{n^{7c}} - \frac{1}{2^{n^\alpha}} \quad \text{(Equations 7.2, 7.3 \& sampling error)} \\
& \geq \; \epsilon(n) + \frac{1}{n^{3c}}
\end{aligned}
$$

Notice that the second inequality follows from constraints on the probability that a random function $w \in \mathcal{F}^n$ is in the set S^n defined in equation 7.2. This is because, for all sufficiently large n, K'_n is greater than the value of K_n. As previously seen, the probability that a random $w \in \mathcal{F}^n$ has the property that $\Pr_{C_n}(\mathsf{G}^n \diamond w) > K_n$, is the probability that w is contained in the set S^n. But, we know that the probability of a random $w \in \mathcal{F}^n$ being in the set S^n is less than $\frac{1}{n^{7c}}$. Clearly it follows that the probability that a random $w \in \mathcal{F}^n$ has the property that $\Pr_{C_n}(\mathsf{G}^n \diamond w) > K'_n$ is smaller than $\frac{1}{n^{7c}}$, for the set of functions with this property is necessarily a subset of S^n. Clearly similar arguments apply for when $\Pr_{C_n}(\mathsf{G}^n \diamond w) < L_n$.

Finally, since $B_n(w)$ approximates $\Pr_{C_n}(\mathsf{G}^n \circ w)$ to within a value of $\frac{1}{2^{n^\alpha}}$, for all but $\frac{1}{2^{n^\alpha}}$ of the $w \in \mathcal{F}^n$, and since it still holds that for all sufficiently large n that $K'_n - \frac{1}{2^{n^\alpha}}$ is greater than K_n, there can only be a fraction $\frac{2}{n^{7c}} + \frac{1}{2^{n^\alpha}}$ of the $w \in \mathcal{F}^n$ for which $B'(w)$ accepts. The first $\frac{2}{n^{7c}}$ of these functions are the ones in S^n and T_n. The last $\frac{1}{2^{n^\alpha}}$ of these functions are the functions w in which $B(w)$'s approximation of $\Pr_{C_n}(\mathsf{G}^n \diamond w)$ is off by more than a factor of $\frac{1}{n^{7c}}$.

We let $\Upsilon_n = \bar{B}_n$, and we are finished.

7.3 Questioning The Model

In this section we have shown how to construct a PRFG from partially secure PRFGs, and have thus reached one of our research goals. However, the initial purpose of studying constructions which increased the security of partially secure PRFGs was to attempt to provide evidence that such constructions would amplify security when applied to completely insecure function generators (what would be considered a 0-secure PRFG in our model). Or in lay man's terms, we wanted to provide evidence that there was reason to believe that the constructions used in real world cryptographic permutation generators, such as DES and the AES candidates, actually do increase security. Unfortunately, our research program has fallen short in this respect because there is little reason to believe that the construction studied in this chapter would have any security amplifying properties on insecure function generators. This is based on the belief that, to the best of the authors' knowledge, there exists no "real world" cryptographic system, in public use, that is based on the security amplyfiyng properties of the exlcusive-or function. Further, the author predicts that the consensus among cryptologists would be that it would be shocking to discover that the exclusive-or function can be used in the fashion described in this chapter to increase security. Therefore, while there is no formal evidence backing the claim that the $\diamond$ operator does not provide security amplification in the "real-world", there is sufficient enough evidence to have us question the model we are working in. Therefore, proving that certain constructions are security increasing when applied to partially secure PRFGs may provide little evidence that the construction will have security amplifying properties when applied to trivial generators.

Chapter 8

Conclusions and Open Questions

In this chapter we summarize our findings, and propose some interesting new research directions.

8.1 Conclusions

Our goal was to understand in which ways insecure functions generators might be combined in order to produce function generators which are conjectured to be pseudo-random. We developed a list of possible constructions which are based on "natural" operators which are commonly used in the cryptographic community.

We then considered each construction in the model of partially secure pseudo-random generators. For each proposed construction we were then able to either give evidence which suggests it is of little practical use, or classify the construction as security increasing, preserving, or decreasing. We extended the result of Luby and Rackoff [12] and showed that we could achieve a $1 - \frac{1}{\log n}$ secure generator from a partially secure PRPG. We then showed that the we could achieve the same security using PRFGs and the $\Box$ operator, as proposed by Luby and Rackoff [13]. Finally, we have shown that we can achieve a completely secure PRFG from a partially secure PRFG based on the $\Diamond$ operator generator. This last result has led us to question the belief that a construction which is security increasing in the weakly secure pseudo-random model is also security increasing when applied to insecure generators. However, it has answered in the positive the question of whether or not it is possible to use a "natural" construction to combine partially secure PRFGs into a secure PRFG.

8.2 Open Questions

In this section we will first review the open questions which have appeared previously in this text. We will then describe one problem which we have spent a significant amount of time thinking about, but for which we have made little progress on. Finally, we will suggest a new direction for future research which we believe is of strong practical importance, and if successful may dramatically change the way conjectured PRPGs and PRFGs are designed.

8.2.1 Previously Stated Open Questions

We remind the reader that in Chapter 4 we could not show that the construction based on the XOR operator was security preserving when applied to generators of security greater than $\frac{1}{2}$. We then mentioned that for similar reasons we were unable to determine whether a construction based on the $\Box$ and XOR operators was security increasing, preserving, or possibly diminishing. We restate the questions here, for completeness.

Open Question 8.1 (Restatement of Open Question 4.1) *Is the construction* $F(\alpha) = (\mathsf{G}_{m(n)} \oplus \cdots \oplus \mathsf{G}_1)(\alpha)$ *security increasing, when the generators* G_i *are restricted to being* $1 - \delta$ *secure for* $\delta < \frac{1}{2}$.

Open Question 8.2 (Restatement of Open Question 4.2) *Is the construction* $\mathsf{F}(\alpha) = \left((\mathsf{G}_{2m(n)} \Box \mathsf{G}_{2m(n)-1}) \oplus \cdots \oplus (\mathsf{G}_2 \Box \mathsf{G}_1)\right)(\alpha)$ *security increasing, security preserving or security decreasing.*

8.2.2 Adaptive vs. Non-Adaptive Security

We say that a PRFG is non-adaptively secure if it is secure under a weakened version of the standard definition of secure PRFGs. The definition is weakened by disallowing the adversary from making adaptive queries to the function-oracle. Therefore, the adversary can still make a polynomial in n number of queries, but the queries must all be made at the same time. This prevents the adversary from making queries based on information learned from the results of previous queries.

We give a simple example of a generator which is non-adaptively secure, but which is adaptively insecure. Let $\widehat{\mathsf{G}} = \{\widehat{\mathsf{G}^n} : \{0,1\}^{\kappa(n)} \times \{0,1\}^n \rightarrow \{0,1\}^n | n \in \mathbb{N}\}$ be an adaptively secure PRFG. We construct a non-adaptively secure generator $\mathsf{G} = \{\mathsf{G} : \{0,1\}^{\kappa(n)} \times \{0,1\}^n \rightarrow \{0,1\}^n | n \in \mathbb{N}\}$, by taking for each $k \in \{0,1\}^{\kappa(n)}$ the function $\hat{g}_k^n \in \widehat{\mathsf{G}^n}$, and using it to construct a new function $g_k^n \in \mathsf{G}^n$: for all $x \neq \hat{g}_k^n(\bar{0})$, for some $x \in \{0,1\}^n$ and $k \in \{0,1\}^{\kappa(n)}$ we set $g_k^n(x) = \hat{g}_k^n(x)$, but we set $g_k^n(\hat{g}_k^n(\bar{0})) = \bar{0}$.

The question is if there is some construction based on "natural" operators which allows us construct an adaptively secure PRFG, from non-adaptively secure generators. Specifically, we say that a construction provides adaptive security if, for all possible non-adaptively secure PRFGs used in the construction, the resulting generator is adaptively secure.

We note that using a non-adaptively secure PRFG we can easily construct a one-way function. Then using the standard constructions we can construct an adaptively secure PRFG. Therefore, when we limit our operators and constructions to be natural, we mean that the operators are ones which can easily be used to combine generators. Further, the operators are normally computable in linear time.

As we just mentioned, the existence of non-adaptively secure PRFGs implies the existence of adaptively secure PRFGs. Therefore, the point of this question is to develop an efficient construction which gives adaptively secure generators from non-adaptive ones.

We will now rule out one feasible construction, and propose several others.

Data Dependent Re-Keying Does Not Work

Let $\mathsf{H} = \{\mathsf{H}^n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n | n \in \mathbb{N}\}$ and $\mathsf{G} = \{\mathsf{G}^n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n | n \in \mathbb{N}\}$ be non-adaptively secure PRFGs. Let the data-dependent re-keying construction $\mathsf{F} = \{F^n | \in \mathbb{N}\}$ be defined as follows:

$$\mathsf{F}^n(k,x) = \mathsf{G}^n(\mathsf{H}^n(k,x),x),$$

for each n.

We will now construct specific generators G^n and H^n which are non-adaptively secure, but F will not be adaptively secure. Let $\widehat{\mathsf{H}} = \{\widehat{\mathsf{H}}^n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n | n \in \mathbb{N}\}$ and $\widehat{\mathsf{G}} = \{\widehat{\mathsf{G}}^n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n | n \in \mathbb{N}\}$ be adaptively secure PRFGs. For each n we construct G^n by setting for each $x \in \{0,1\}^n$ and each $k \in \{0,1\}^n \setminus \{\bar{0}\}$ the function $g_k^n(x) = \hat{g}_k^n(x)$; and then for each $x \in \{0,1\}^n$ setting $g_{\bar{0}}^n(x) = \bar{0}$. For each n and each $k \in \{0,1\}^n$ let $\alpha_k^n = g^n_{\hat{h}_k^n(\bar{0})}(\bar{0})$. For each n we construct H^n by setting for each $k \in \{0,1\}^n$ and each $x \in \{0,1\}^n \setminus \{\alpha_k^n\}$ the function $h_k^n(x) = \hat{h}_k^n$, and setting $h_k^n(\alpha_k^n) = \bar{0}$. Notice that G^n is adaptively secure, and that H^n is non-adaptively secure.

Now construct F from H and G, as in the data-dependent re-keying construction given above. Notice that for each n and for each $k \in \{0,1\}^n$ the function $f_k^n \in \mathsf{F}^n$ has the property that $f_k^n(f_k^n(\bar{0})) = \bar{0}$, and therefore the generator F is clearly not adaptively secure. Further, since G is adaptively secure, we see that the construction does not even preserve adaptive security.

Proposed Constructions

We conjecture that any of the constructions presented in Chapter 4 which are security improving might also provide adaptive security. In particular, we have not been able to give examples of constructions of any non-adaptively secure PRFGs which, when used in one of the security improving constructions of Chapter 4, gives a generator which is not adaptively secure. However, we have failed to give either a proof of adaptive security or a counterexample when we restrict the constructions so that only two constituent non-adaptively secure generators are used, and when only one adaptive query is performed.

We also conjecture that if K is a 1-wise independent permutation generator, and G is a non-adaptively secure pseudo-random permutation generator then the construction $\mathsf{G} \circ \mathsf{K}$ is an adaptively secure pseudo-random permutation generator. We note that composition is not commutative, and the construction $\mathsf{K} \circ \mathsf{G}$ is not conjectured to provide adaptive security, for there exist specific examples of pair-wise independent permutation generators and non-adaptively secure PRFGs which when used in the above construction do not give adaptive security.

For example if we let $\mathsf{K} = \{\mathsf{K}^n : \{0,1\}^{2n} \times \{0,1\}^n \rightarrow \{0,1\}^n | n \in \mathbb{N}\}$ be a pair-wise independent permutation generator, where $\mathsf{K}(a \bullet b, x) = ax + b$; $|a| = |b| = n$; $a \neq \bar{0}$; and the operations are performed over the field $GF(2^n)$. Let $\mathsf{H} = \{\mathsf{H}^n : \{0,1\}^{\kappa(n)} \times \{0,1\}^n \rightarrow \{0,1\}^n | n \in \mathbb{N}\}$ be an adaptively secure PRPG. For each n, and for each $k \in \{0,1\}^{\kappa(n)}$, we will modify $h_k^n \in \mathsf{H}$ and thereby make H non-adaptively secure. Let $\alpha_{h_k^n} = \frac{h_k^n(0) - h_k^n(1)}{h_k^n(0) - h_k^n(2)}$. Modify h_k^n by making the changes:$h_k^n(\alpha_{h_k^n}) \leftarrow h_k^n(0)$, $h_k^n(\alpha_{h_k^n} + 1) \leftarrow h_k^n(1)$ and $h_k^n(\alpha_{h_k^n} + 2) \leftarrow h_k^n(2)$. Clearly, H^n is no longer adaptively secure as one can easily construct an adversary which accepts a function $f^n \in \mathcal{F}^n$ iff $f^n(0) = f^n(\alpha_{f^n})$, $f^n(1) = f^n(\alpha_{f^n} + 1)$ and $f^n(2) = f^n(\alpha_{f^n} + 2)$. However, we can also construct an adversary which distinguishes $\mathsf{K} \circ \mathsf{H}$ adaptively. Given a function $f^n \in \mathcal{F}^n$ our adversary will distinguish it by performing computing α_{f^n}, and then computing $\beta_{f^n} = \frac{f^n(\alpha_{f^n}) - f(\alpha_{f^n}+1)}{f(\alpha_{f^n}) - f(\alpha_{f^n}+2)}$. The adversary accepts iff $\beta_{f^n} = \alpha_{f^n}$. It is a relatively simple argument to show that this will occur with probability 1 if f^n was drawn from $\mathsf{K}^n \circ \mathsf{H}^n$, whereas it will occur with a negligible probability if f was drawn from $\mathcal{F}^n$. This construction can be generalized to show that for any k-wise independent *function* generator $\mathsf{G} = \{\mathsf{G}^n : \{0,1\}^{kn} \times \{0,1\}^n \rightarrow \{0,1\}^n : n \in \mathbb{N}\}$, where $g^n_{a_k \circ \cdots \circ a_1}(x) = a_k x^{k-1} + a_{k-1} x^{k-2} + \cdots + a_1 x^0$ and $|a_i| = |x| = n$; and a generator H' which is constructed in a generalized manner of the generator H presented above, that the generator $\mathsf{G} \circ \mathsf{H}'$ is non-adaptively secure.

8.2.3 A Combinatorial Security Model

As mentioned earlier, the existence of PRFGs cannot be proven without indirectly proving that $\mathcal{P} \neq \mathcal{NP}$. Given the apparent difficulty of such a task, it would seem unlikely that any modern cryptographic systems will be proven secure in the immediate future. However, this does not prevent the development of a theory which can be used to help cryptographers design cryptographic systems. Currently, block-cipher designers only seem to have intuition to guide their designs. Further, the number of designs often seem to be as numerous as the number of block-cipher designers. The result is that there are many proposed block-ciphers, none of which can be compared. We propose the development of a theory to help with this process.

This research direction is motivated by the current Advanced Encryption Standard competition. As mentioned earlier, there are many proposed function and permutation generators which are currently being considered, and there is no quantifiable method for comparing the different designs. The end result seems to be that the main criteria for choosing a winner of this competition is based on the pedigree, within the cryptographic community, of the designers. Yet many of these designs are based on composing trivial permutation generators numerous times to result in a permutation generator which appears to be secure. A theory such as the one we are proposing would allow for these generators to be compared in a quantitative, and hopefully meaningful sense. Information about the AES competition and the currently proposed ciphers is available at the AES' official web page `http://csrc.nist.gov/encryption/aes/aes_home.htm`.

The idea behind our proposed theory would be to combine function generators which have some trivial security properties together in some construction, such as the security increasing constructions proposed in Chapter 4. We would then conjecture that if the resulting generator has some specific combinatorial property, then it is a pseudo-random generator.

An example of how such a theory might look is the following. We would like to stress that these examples are most likely incorrect, and are included only to help bring light to our proposed research direction.

Definition 8.1 (Trivial Permutation Generator (Example)) *A function generator* $\mathsf{T} = \{\mathsf{T}^n : \{0,1\}^{\kappa(n)} \times \{0,1\}^n \to \{0,1\}^n | n \in \mathbb{N}\}$ *is considered trivial, if for each* n*, each key* $k \in \{0,1\}^{\kappa(n)}$ *and each* $x \in \{0,1\}^n$*, every bit of* $t_k^n(x)$ *is dependent on no more than* $\mathcal{O}(\log n)$ *bits of* x*.*

An example of a construction which would be used to combine trivial function generators is repeated composition. Finally, an example of a conjecture of security based on a combinatorial principle is given below.

Conjecture 8.1 (Combinatorial Security (Example)) *Let* T *be a trivial function generator. Construct* $\mathsf{F} = \mathsf{T} \circ \cdots \circ \mathsf{T}$*, so that enough compositions are performed so that* F *is an almost pair-wise independent function generator. Then there exists a polynomial* p *such that* $\widehat{\mathsf{F}} = \{\underbrace{\mathsf{F}^n \circ \cdots \circ \mathsf{F}^n}_{p(n)} | n \in \mathbb{N}\}$ *is a PRFG.*

Note the importance of T being a trivial function generator, for there are standard constructions of pair-wise independent permutation generators which are clearly not secure. For example, if we consider the generator $\mathsf{K}(a \bullet b, x) = ax + b$, where $|x| = |a| = |b| = n$, $a \neq \bar{0}$ and the operations are performed over the field $GF(2^n)$, then it is clearly insecure. Further, this generator would be insecure under an arbitrary number of compositions, as the resulting permutations will always be linear. Therefore, the triviality requirement would exclude the use of a generator such as K.

Notice that this construction is based on what people currently do to construct ciphers, and is not some notion we have invented. Both DES and almost all of the AES candidates can by described at an abstract level as being the result of composing together many completely insecure (or trivial) permutation generators. What differs in their design is the types of trivial permutation generators they use, and how many times the generators are composed. Notice, that is exactly the types of differences which could be compared in our proposed theory.

One difference between our proposed theory, and the construction of known ciphers, such as DES and the AES candidates, is that the known ciphers are compositions of permutation generators, whereas in our construction we consider compositions of function generators. Initially, we thought that the trivial generators should be restricted to being permutation generators. However, in hindsight we realize that this restriction is probably not important. The important goal is that we are able to achieve an almost pair-wise independent function generator from some polynomial number of compositions.

We believe that using a theory such as the one proposed, designers could easily quantify and compare tradeoffs which were made during design. For example, arguments could be made for using different types of initial trivial generators. However, we could now argue tradeoffs based on how fast specific trivial generators are, and the size of the construction in which they would need to be embedded, in order for the construction to be conjectured secure. Further, it might be possible to prove security preserving reductions between different generators in this model. This would allow us to establish security classes, where certain generators would be pseudo-random only if other generators are pseudo-random. Finally, because the theory is not based on any complexity theoretic assumptions, it may be easier than in the complexity theoretic models to find counter examples, and thus disprove the theory, if the theory turned out to be incorrect.

Bibliography

[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 *Lecture Notes In Computer Science*. Springer-Verlag, 1998.

[2] K. Akcoglu and C. Rackoff. Pseudo-random permutation generators and cryptographic composition. (Manuscript in preperation for release)., June 1998.

[3] E. Biham and A. Shamir. *Differential Cryptanalysis of the Data Encryption Standard*. Springer-Verlag, 1993.

[4] T. Cormen, C.E. Lierson, and R. Rivest. *Introduction to Algorithms*. MIT Press and McGraw Hill, 1990.

[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] Oded Goldreich. *Foundations of Cryptography(Fragments of a Book)*. Weizmann Institute of Science, 1995.

[8] R. Impagliazzo. Hard core distributions for somewhat hard problems. `http://www-cse.ucsd.edu/~russell/`, 1994.

[9] J. Kilian and P. Rogaway. How to protect DES against exhaustive key search. `http://wwwcsif.cs.ucdavis.edu/~rogaway`, 1997.

[10] J. Killian and P. Rogaway. How to protect DES against exhaustive key search. In *Advances in Cryptology – Crypto '96*, Lecture Notes In Computer Science, 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 18th Annual Symposium on Theory of Computing*. ACM, 1986.

[13] M. Luby and C. Rackoff. Secure cryptography from a slightly secure function generator. (Private notes on cryptography)., October 1987.

[14] M. Luby and C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal on Computing*, 17:373–386, 1988.

[15] Micheal Luby. *Pseudorandomness and Cryptographic Applications*. Princeton Computer Science Notes. Pinceton University Press, 1996.

[16] M. Matsui. Linear cryptanalysis of DES cipher (i). In *Advances in Cryptology – EUROCRYPT 93 Proceedings*, LNCS, pages 386–397. Springer-Verlag, 1994.

[17] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[18] M. Naor and O. Reigold. On the construction of psedo-random permutations: Luby-rackoff revisited. In *Proceedings of the 29th Annual ACM Symposium on the Theory of Computing*, 1997. To appear in the Journal of Cryptology.

[19] Christos Papadimitriou. *Computational Complexity*. Addison-Wesley Publishing Company, 1994.

[20] P. Rogaway. *The Security of DESX.* Cryptobytes, 1996.

[21] Andrew Yao. Theory and applications oif trapdoor functions (extended abstract). In *Proceedings of the 23rd Symposium on Foundations of Computer Science.* IEEE, 1982.

Appendix A

Proof Of Security for the Generator Described in 5.1.1

In this section we present the proof of security for the PRPG constructed in section 5.1.1. We first remind the reader of the construction presented, and then give the proof of its security.

A.1 Construction of the $1-\delta$ Secure Generator

To simplify the presentation we assume that δ and ϵ are of the form $\frac{1}{2^c}$ or $1-\frac{1}{2^c}$, for some constant c. Let $\widehat{\mathsf{G}} = \{\widehat{\mathsf{G}}^n : \{0,1\}^{\ell(n)} \times \{0,1\}^n \to \{0,1\}^n | n \in \mathbb{N}\}$ be a PRPG. We construct $\mathsf{G} = \{\mathsf{G}^n : \{0,1\}^{\ell(n)+c} \times \{0,1\}^n \to \{0,1\}^n | n \in \mathbb{N}\}$ to be $1-\delta$ secure in two steps. We will describe the construction of G^n, and note that the construction of H^n is similar.

First we set a fraction δ of the keys to correspond to the identity permutation, and the remainder to correspond to permutations chosen from $\widehat{\mathsf{G}}^n$. This is done in two different fashions dependent on the form of δ as described below:

Case 1 ($\delta = 1-\frac{1}{2^c}$) For each $k \in \{0,1\}^{\ell(n)+c}$ we set the permutation $g_k^n \in \mathsf{G}^n$ to be the identity permutation if the first c bits of k *are not* all 0; otherwise let $\bar{k}$ be the last $\ell(n)$ bits of k, and set $g_k^n = \hat{g}_{\bar{k}}^n$, for $\hat{g}_{\bar{k}}^n \in \widehat{\mathsf{G}}^n$.

Case 2 ($\delta = \frac{1}{2^c}$) For each $k \in \{0,1\}^{\ell(n)+c}$ we set the permutation $g_k^n \in \mathsf{G}^n$ to be the identity permutation if the first c bits of k *are* all 0; otherwise let $\bar{k}$ be the last $\ell(n)$ bits of k, and set $g_k^n = \hat{g}_{\bar{k}}^n$, for $\hat{g}_{\bar{k}}^n \in \widehat{\mathsf{G}}^n$.

The second step in the construction of G^n is to ensure that for each $k \in \{0,1\}^{\ell(n)+c}$ we set the value of $g_k^n(\bar{0})$ in one of two manners, depending on the form of δ. We describe this transformation below:

Case 1 ($\delta = 1 - \frac{1}{2^c}$) For all k set the first c bits of $g_k^n(\bar{0})$ to 0; the last $n-c$ bits remain as they were in $g_k^n(\bar{0})$.

Case 2 ($\delta = \frac{1}{2^c}$) For all k ensure that not all of the first c bits of $g_k^n(\bar{0})$ are 1. If they are, set them to be a member of the set $\{0,1\}^c \setminus \{1^c\}$ chosen uniformly at random[1]; the last $n-c$ bits remain as they were in $g_k^n(\bar{0})$.

Notice that in order to maintain the permutation property of G^n we can simply store both the initial and modified value of $g_k^n(\bar{0})$. Should a query ever be made to the initial preimage of the modified value of $g_k^n(\bar{0})$, then we respond with the initial value of $g_k^n(\bar{0})$.

A.2 Proof that G is $1-\delta$ Secure

Theorem A.1 *The pseudo-random permutation generator* G *is* $1-\delta$ *secure.*

Proof: Let A be an adversary such that:

$$\left| \Pr_{f^n \in \mathcal{F}^n}(A^{f^n} = 1) - \Pr_{g^n \in \mathsf{G}^n}(A^{g^n} = 1) \right| \geq \delta + \frac{1}{n^d}, \tag{A.1}$$

[1]In practice a PRNG p would be used to set the first c bits of $g_k^n(\bar{0})$. This would be done by setting the first c bits of $g_k^n(\bar{0})$ to be equivalent be the first consecutive set of c bits in $p(k)$ which were not all 1. This allows the first c bits to be computed, and discards the need to store them, which would not be possible as it would require an exponential amount of storage.

for infinitely many n and some constant $c > 0$. We will show how to distinguish $\widehat{\mathsf{G}}$ from random functions or derive a contradiction, and thereby prove the $1-\delta$ security of G. The intuition is as follows, we expect the adversary, A, to be able to distinguish the identity permutations in G^n from random functions in $\mathcal{F}^n$. However, the remaining functions in G^n should not be distinguishable from random functions $f^n \in \mathcal{F}^n$, which have the property that $f^n(\bar{0})$'s first c bits are either all 0, or not all 1, dependent on whether $\delta \geq \frac{1}{2}$ or $\delta < \frac{1}{2}$ respectively. Further, accepting any of the $f^n \in \mathcal{F}^n$ which do not conform to the above description will only decrease the adversaries distinguishing probability. Therefore, it would appear that the distinguishing probability should not be better than δ, and this provides the contradiction.

We partition the set $\{0,1\}^n$ into two sets X^n and $Y^n = \{0,1\}^n \setminus X^n$. The contents of the set X^n is dependent on the form of δ as described below:

Case 1 $(\delta = 1 - \frac{1}{2^c})$ Then $X^n = \{x \in \{0,1\}^n | \text{The first } c \text{ bits of } x \text{ are } 0\}$.

Case 2 $(\delta = \frac{1}{2^c})$ Then $X^n = \{x \in \{0,1\}^n | \text{The first } c \text{ bits of } x \text{ are not all } 1\}$.

We use X^n and Y^n to distinguish between the functions f^n which appear to have had the transformation on $f^n(\bar{0})$ applied to them, in which case $f^n(\bar{0}) \in X^n$, versus those which have not, in which case $f^n(\bar{0}) \in Y^n$. Finally we distinguish the identity permutations from random permutations by simply checking of $f^n(\bar{0}) = \bar{0}$. By partitioning the functions in the distributions as in the manner described above, and assuming that $\Pr_{f^n \in \mathcal{F}^n}(A^{f^n} = 1) > \Pr_{g^n \in \mathsf{G}^n}(A^{g^n} = 1)$ (if this is not the case, then we can simply reverse the outputs of A), we can rewrite equation (A.1) as shown below:

$$
\begin{aligned}
& \Pr_{f^n \in \mathcal{F}^n}[A^{f^n} = 1 | f^n(\bar{0}) = \bar{0}] \Pr_{f^n \in \mathcal{F}^n}[f^n(\bar{0}) = \bar{0}] \\
+ \; & \Pr_{f^n \in \mathcal{F}^n}[A^{f^n} = 1 | f^n(\bar{0}) \neq \bar{0} \;\&\; f^n(\bar{0}) \in X^n] \Pr_{f^n \in \mathcal{F}^n}[f^n(\bar{0}) \neq \bar{0} \;\&\; f^n(\bar{0}) \in X^n] \\
+ \; & \Pr_{f^n \in \mathcal{F}^n}[A^{f^n} = 1 | f^n(\bar{0}) \neq \bar{0} \;\&\; f^n(\bar{0}) \in Y^n] \Pr_{f^n \in \mathcal{F}^n}[f^n(\bar{0}) \neq \bar{0} \;\&\; f^n(\bar{0}) \in Y^n]
\end{aligned}
$$

$$
\begin{aligned}
- & \Pr_{g^n \in \mathsf{G}^n}[A^{g^n} = 1 | g^n(\bar{0}) = \bar{0}] \Pr_{g^n \in \mathsf{G}^n}[g^n(\bar{0}) = \bar{0}] \\
- & \Pr_{g^n \in \mathsf{G}^n}[A^{g^n} = 1 | g^n(\bar{0}) \neq \bar{0} \,\&\, g^n(\bar{0}) \in X^n] \Pr_{g^n \in \mathsf{G}^n}[g^n(\bar{0}) \neq \bar{0} \,\&\, g^n(\bar{0}) \in X^n] \\
- & \Pr_{g^n \in \mathsf{G}^n}[A^{g^n} = 1 | g^n(\bar{0}) \neq \bar{0} \,\&\, g^n(\bar{0}) \in Y^n] \Pr_{g^n \in \mathsf{G}^n}[g^n(\bar{0}) \neq \bar{0} \,\&\, g^n(\bar{0}) \in Y^n] \\
\geq & \delta + \frac{1}{n^d}
\end{aligned}
$$

We notice that for the set of random functions it is the case that:

$$
\begin{aligned}
& \Pr_{f^n \in \mathcal{F}^n}[f^n(\bar{0}) = \bar{0}] = \frac{1}{2^n}; \\
& \Pr_{f^n \in \mathcal{F}^n}[f^n(\bar{0}) \neq \bar{0} \,\&\, f^n(\bar{0}) \in X^n] = 1 - \delta - \frac{1}{2^n}; \\
& \Pr_{f^n \in \mathcal{F}^n}[f^n(\bar{0}) \neq \bar{0} \,\&\, f^n(\bar{0}) \in Y^n] = \delta.
\end{aligned}
$$

Similarly, by the construction of G^n we know that

$$
\begin{aligned}
& \Pr_{g^n \in \mathsf{G}^n}[g^n(\bar{0}) = \bar{0}] = (\delta + \eta(n)); \\
& \Pr_{g^n \in \mathsf{G}^n}[g^n(\bar{0}) \neq \bar{0} \,\&\, g^n(\bar{0}) \in X^n] = 1 - \delta - \eta(n); \\
& \Pr_{g^n \in \mathsf{G}^n}[g^n(\bar{0}) \neq \bar{0} \,\&\, g^n(\bar{0}) \in Y^n] = 0;
\end{aligned}
$$

for some negligible function $\eta(n)$. The quantity $\eta(n)$ comes from the observation that $\Pr_{\hat{g}^n \in \widehat{\mathsf{G}}}(\hat{g}^n(\bar{0}) = \bar{0})$ must be at most $\eta(n)$, or $\widehat{\mathsf{G}}$ is not pseudo-random.

Therefore, rewriting equation (A.1) again, using the above facts, we get:

$$
\begin{aligned}
& \Pr_{f^n \in \mathcal{F}^n}[A^{f^n} = 1 | f^n(\bar{0}) = \bar{0}] \left(\frac{1}{2^n}\right) \\
+ & \Pr_{f^n \in \mathcal{F}^n}[A^{f^n} = 1 | f^n(\bar{0}) \neq \bar{0} \,\&\, f^n(\bar{0}) \in X^n] \left(1 - \delta - \frac{1}{2^n}\right) \\
+ & \Pr_{f^n \in \mathcal{F}^n}[A^{f^n} = 1 | f^n(\bar{0}) \neq \bar{0} \,\&\, f^n(\bar{0}) \in Y^n] \delta \\
- & \Pr_{g^n \in \mathsf{G}^n}[A^{g^n} = 1 | g^n(\bar{0}) = \bar{0}] \delta + \eta \\
- & \Pr_{g^n \in \mathsf{G}^n}[A^{g^n} = 1 | g^n(\bar{0}) \neq \bar{0} \,\&\, g^n(\bar{0}) \in X^n] (1 - \delta - \eta) \\
\geq & \delta + \frac{1}{n^d}
\end{aligned} \tag{A.2}
$$

We now use a lemma which formalizes our intuition that the adversaries should not be able to distinguish between random functions $f^n \in \mathcal{F}^n$ that have the property that $f^n(\bar{0}) \in X^n$ and those $g^n \in \mathsf{G}^n$ which are not the identity function. These have the form $g^n(\bar{0}) \in X^n$, by the construction of G^n.

Lemma A.1 *For all constants $e \geq 0$ and sufficiently large n,*

$$\left| \Pr_{f^n \in \mathcal{F}^n}[A^{f^n} = 1 | f^n(\bar{0}) \neq \bar{0} \ \& \ f^n(\bar{0}) \in X^n] - \Pr_{g^n \in \mathsf{G}^n}[A^{g^n} = 1 | g^n(\bar{0}) \neq \bar{0} \ \& \ g^n(\bar{0}) \in X^n] \right| < \frac{1}{n^e}.$$

We now take equation (A.2) and eliminate all negligible quantities. Further, we treat the two probabilities in Lemma A.1 as equal because their difference is negligible. The result is:

$$\Pr_{f^n \in \mathcal{F}^n}[A^{f^n} = 1 | f^n(\bar{0}) \neq \bar{0} \ \& \ f^n(\bar{0}) \in Y^n]\delta - \Pr_{g^n \in \mathsf{G}^n}[A^{g^n} = 1 | g^n(\bar{0}) = \bar{0}]\delta \geq \delta + \frac{1}{n^d}.$$

This gives:

$$\Pr_{f^n \in \mathcal{F}^n}[A^{f^n} = 1 | f^n(\bar{0}) \neq \bar{0} \ \& \ f^n(\bar{0}) \in Y^n] - \Pr_{g^n \in \mathsf{G}^n}[A^{g^n} = 1 | g^n(\bar{0}) = \bar{0}] \geq 1 + \frac{1}{\delta n^d},$$

but this is a contradiction since the two probabilities are bounded to be between 0 and 1. ...$\square$

All that remains is to give a proof of Lemma A.1.

Proof of Lemma A.1: Suppose there exists a e such that for infinitely many n the inequality stated in the lemma does not hold. We will construct an adversary $\widehat{A}$ which can distinguish $\widehat{\mathsf{G}}^n$ from $\mathcal{F}^n$ with significant probability for infinitely many n.

Given a function w, $\widehat{A}$ will query $w(\bar{0})$, and if $w(\bar{0}) = \bar{0}$ then $\widehat{A}$ will accept with probability $\frac{1}{2}$. Otherwise $\widehat{A}$ simulates A running on w exactly, except in one condition: if A queries $w(\bar{0})$ and if $w(\bar{0}) \notin X^n$ then we modify it, dependent on δ.

Case 1 ($\delta = 1 - \frac{1}{2^c}$) Set the first c bits of $w(\bar{0})$ to $\bar{0}$.

Case 2 ($\delta = \frac{1}{2^c}$) Set the first c bits of $w(\bar{0})$ to be a member of the set $\{0,1\}^c \setminus \{1^c\}$ chosen uniformly at random.

We maintain the permutation property of w by simply having $\widehat{A}$ store both the initial and modified value of $w(\bar{0})$. Should a query ever be made to the initial preimage of the modified value of $w(\bar{0})$, then $\widehat{A}$ responds with the initial value of $w(\bar{0})$. Finally, $\widehat{A}$ accepts *iff* A accepts.

We notice that the probability of the event in which $w(\bar{0}) = \bar{0}$ can be bounded in both distributions. For the set of random functions, $\Pr_{f \in \mathcal{F}^n}(f(\bar{0}) = \bar{0}) = \frac{1}{2^n}$, and for the functions from the generator $\widehat{\mathsf{G}}^n$, $\Pr_{g^n \in \widehat{\mathsf{G}}^n}(g^n(\bar{0}) = \bar{0}) \leq \frac{1}{n^e}$, for all constants $e > 0$. Since in both cases the event happens with negligible probability, the fact that $\widehat{A}$ accepts in both these cases with probability $\frac{1}{2}$, does not have any significant affect on $\widehat{A}$'s distinguishing probability on the two distributions of functions. Therefore, we will ignore this factor for the remainder of the proof.

We observe that by the simulation of A by $\widehat{A}$ and the construction of G from $\widehat{\mathsf{G}}$ that:

$$\Pr_{g^n \in \mathsf{G}^n}[A^{g^n} = 1 | g^n(\bar{0}) \neq \bar{0} \;\&\; g^n(\bar{0}) \in X^n] = \Pr_{\hat{g}^n \in \widehat{\mathsf{G}}^n}[\widehat{A}^{\hat{g}^n} = 1].$$

Similarly,

$$\Pr_{f^n \in \mathcal{F}^n}[A^{f^n} = 1 | f^n(\bar{0}) \neq \bar{0} \;\&\; f^n(\bar{0}) \in X^n] = \Pr_{f^n \in \mathcal{F}^n}[\widehat{A}^{f^n} = 1].$$

Therefore by assumption:

$$\left| \Pr_{\hat{g}^n \in \widehat{\mathsf{G}}^n}[\widehat{A}^{\hat{g}^n} = 1] - \Pr_{f^n \in \mathcal{F}^n}[\widehat{A}^{f^n} = 1] \right| \geq \frac{1}{n^e}.$$

This contradicts the assumption that $\widehat{\mathsf{G}}$ is a PRPG, and proves the claim. ...$\square$