UNIVERSITY OF CALIFORNIA SAN DIEGO

Analyzing the Security Gap in Bootstrapping Obfuscation

A Thesis submitted in partial satisfaction of the
requirements for the degree of Master of Science

in

Computer Science

by

Evan Thomas Roncevich

Committee in charge:

Professor Daniele Micciancio, Chair
Professor Mihir Bellare
Professor Deian Stefan

2018

The Thesis of Evan Thomas Roncevich is approved and is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California San Diego

2018

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

I would like to acknowledge Professor Daniele Micciancio for his support as my professor and the chair of my committee. Through three courses and numerous one-on-one meetings, I've had the chance to improve and really enjoy being a part of the academic community.

I would additionally like to acknowledge the wonderful graduate students at UCSD who would answer my innumerable questions and continually spark my interest in research.

Chapter 1,2, and 3, in part is currently being prepared for submission for publication of the material. Micciancio, Daniele; Roncevich, Evan. The thesis author was the primary investigator and author of this material.

I would like to thank Sandia National Laboratories for funding my Master's degree, including this research. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. SAND no. SAND2018-6145 T.

ABSTRACT OF THE THESIS

Analyzing the Security Gap in Bootstrapping Obfuscation

by

Evan Thomas Roncevich

Master of Science in Computer Science

University of California San Diego, 2018

Professor Daniele Micciancio, Chair

Indistinguishability obfuscation is an extraordinarily versatile primitive, leading many to search for candidate constructions. Of these candidates, many rely on "bootstrapping" techniques that transform an obfuscator for a small class of circuits into an obfuscator for larger class of circuits. While this technique can be achieved in several manners, examining each shows drawbacks in utility ranging from strong assumptions to exponential loss in security. We specifically examine the construction by Applebaum [1] and the construction by Canetti et al. [11], two very similar constructions, to explain why each is different and if the differences are warranted. We prove that the Applebaum construction is not valid under the indistinguishability obfuscation definition, demonstrating an important difference with the Canetti et al. construction. Additionally, we examine the Garg et al. construction [14] in how it relates to the other constructions.

Chapter 1

Overview of Obfuscation and Bootstrapping

1.1 Introduction

Program obfuscation is a subject that has attracted a lot of attention in cryptography recently. There are many ideal uses for being able to change arbitrary programs into obfuscated programs, where obfuscated programs keep the functionality of the original program while otherwise appear unintelligible. There has been significant work on how this idea of obfuscation can be cryptographically defined along with its relation to other well-known cryptographic definitions.

Throughout this work, we analyze some of these cryptographic definitions along with constructions related to obfuscation.

1.1.1 VBB vs iO

The theoretical study of obfuscation as a cryptographic primitive was initiated by the studies of Barak et al. [4] and Hada [16]. From [4] the authors defined two types of obfuscation, virtual black-box obfuscation (VBB) and indistinguishability obfuscation (iO). These definitions involve taking a program, creating a functionally equivalent obfuscated version of the program, and challenging an adversary to learn some information about the obfuscated program. Virtual black box obfuscation requires an obfuscated program to reveal no more information than what could be obtained using the program as a black-box. The weaker definition of indistinguishability obfuscation requires any two functionally equivalent programs, when obfuscated, to be computationally indistinguishable from each other.

In the Barak et al. paper, the authors demonstrate that the VBB definition is an unachievable definition for general classes of programs. The authors present a circuit family which is impossible to obfuscate under the VBB definition as the very circuit itself can reveal information.

As for the iO definition, research such as [7] shows this form of obfuscation is equivalent to functional encryption. These candidate constructions have generally relied on multi-linear maps, and while many have been broken, with no known impossibility results, indistinguishability obfuscation seems a far more achievable goal, with significant research going into the construction and performance improvements.

1.1.2 Uses of Obfuscation

A reason for interest in obfuscation is the extreme versatility of indistinguishability obfuscation. From iO, an enormous number of cryptographic primitives can be derived. This includes one-way functions, fully homomorphic encryption, zero-knowledge proofs, and functional encryption. Because of the large number of uses, several research efforts have gone into proposing candidate constructions and ways of improving constructions.

One candidate construction in particular proposed by Garg et al. [14] was an iO construction which was limited to obfuscating low depth circuit families from the use of branching programs. In order to obfuscate more general circuit families with polynomial depth, they additionally offer an “amplification” technique which converts a low depth obfuscator into an obfuscator for polynomial-sized circuits. An active area of research is examining how this iO “amplification” technique is performed. We refer to constructions which transform an obfuscator for a small circuit family into an obfuscator for a larger circuit family as obfuscation bootstrapping.

1.1.3 Tradeoffs of Various Methods

A bootstrapping technique (for either the iO or VBB definitions) takes an obfuscator which can securely obfuscate circuit families in some class **WEAK** (such as NC^1) and constructs a new obfuscator which can obfuscate arbitrary polynomial-sized circuit families. There are a number of techniques given for bootstrapping an iO obfuscator in addition to the one proposed by Garg et al. [14].

Garg et al.'s candidate construction for bootstrapping uses Fully Homomorphic Encryption with an obfuscator capable of decrypting in **WEAK** to produce the bootstrapped obfuscator. The first drawback of this technique is to rely on a Fully Homomorphic Encryption scheme which can be implemented in the complexity class **WEAK**. Many FHE schemes rely on the use of the Learning with Errors problem, and significant work has been done in improving the performance of such schemes [13], [12]. This is not an unreasonable assumption to make. There may be additional concerns about performance of FHE techniques and how it would ultimately affect the obfuscation bootstrapping technique, giving interest in relaxing the assumptions needed for an obfuscation bootstrapping construction.

The next construction examined how obfuscation bootstrapping could be accomplished without FHE. In [1], Applebaum examines obfuscation using the VBB definition instead of iO. While it is known that the VBB definition is impossible to achieve for general circuit families, when used to bootstrap under the VBB definition, this construction does not rely on FHE. Instead it uses constructions derived from one-way functions. Applebaum uses a randomized encoding schemes constructable in NC^1 and a family of pseudorandom functions in NC^1 to turn any VBB obfuscator for circuit families in NC^1 into a VBB obfuscator for arbitrary polynomial-sized circuit families. The natural drawback of this construction is the fact that the VBB definition for obfuscation is unachievable for general circuit families as well as needing to rely on the existence of a randomized encoding scheme and a PRF in NC^1.

As the VBB definition is unachievable, Canetti et al. [11], find a way to achieve both obfuscation bootstrapping for iO and avoid relying on FHE for a bootstrapping construction. The authors accomplish this by modifying the construction presented by Applebaum in [1]. This construction still relies on randomized encoding schemes in NC^1, and then restricts the PRF used to a puncturable PRF in NC^1. This construction, however, also has drawbacks, namely an exponential security loss when constructed from the security assumptions of the obfuscator, randomized encoding scheme, and puncturable PRF. In order to prove the security of their bootstrapping construction, the authors rely on an exponential number of security hybrids.

1.1.4 Questions

Through the various construction for bootstrapping obfuscation, several questions come up. The first question is about the exact difference between the constructions of Canetti et al. [11] and Applebaum [1].

The construction by Canetti et al. is extremely similar to the Applebaum construction, making a very specific set of changes to account for the change from VBB to iO definitions. Both use a randomized encoding scheme of a universal circuit to “hide” a circuit that is being evaluated. They both make use of some form of PRF to produce the randomness used by the randomized encoding scheme. The changes however, result in an exponential security loss in the Canetti et al. construction when there was otherwise none.

The first question we seek to answer in this paper is do these constructions need to be different? Is the Applebaum construction sufficient for indistinguishability obfuscation bootstrapping?

If the Applebaum construction is not sufficient for iO while the Canetti et al. construction is sufficient, there must be some gap between the two constructions derived from their differences. Then this leads to the question:

Does the Canetti et al. construction make the most efficient modifications to the Applebaum construction, or are improvements in efficiency possible?

With the analyses of the two constructions, we would also like to compare between the other known forms of bootstrapping.

1.1.5 Contributions

Counterexamples to Applebaum Construction. The main question posed in this work is if the obfuscation bootstrapping construction presented by Applebaum in [1] is secure for indistinguishability obfuscation. Applebaum demonstrates a bootstrapping technique which given a VBB obfuscator for a small class of functions, a family of pseudorandom functions computable in **WEAK** , and a randomized encoding scheme computable in **WEAK** constructs a VBB obfuscator for a larger class of functions. Since VBB and iO have the same syntax/interface, the same construction can be applied to iO. However, the paper makes no claim about its security when using the construction under the indistinguishability obfuscation security definition instead of the VBB definition.

Our first result is to answer this question with "no" in that we find a valid instantiation of Applebaum's construction which trivially fails to bootstrap indistinguishability obfuscation. This first result follows from the specific way Applebaum defines the construction, allowing for the use of the identity function as a trivially secure iO obfuscator for specific circuit families. We present a PRF, randomized encoding scheme, and iO obfuscator that when used in the Applebaum construction, fails the iO definition.

To analyze the Applebaum scheme better adapted to iO we consider how the construction works when requiring nontrivial indistinguishability obfuscation as the base obfuscator and demonstrate that the construction still fails to securely bootstrap iO by demonstrating a general-purpose counterexample to Applebaum's construction. The counterexample relies on constructing a PRF and randomized encoding that when used with any iO obfuscator for the Applebaum construction, fails the iO definition.

From these results we can conclude that Applebaum's construction is suitable for VBB obfuscation but is not sufficient for iO bootstrapping.

Analysis of iO Solutions. Given that Applebaum's construction is not secure for the iO definition, the analysis turns to constructions as found in Canetti et al. [11] or Garg et el. [14]. Each construction is secure but poses tradeoffs in terms of performance and assumption. Because of the similarities found in the Canetti et al. construction and Applebaum construction, we analyze the differences between the schemes to determine how the construction can be improved in terms of efficiency while remaining secure.

Certain parameters in the Canetti et al. construction can be modified while keeping the construction secure. We additionally give a further analysis of the Garg et al. [14] construction to give a more in-depth comparison between these constructions.

1.1.6 Open Question

An important open question worth exploring is the following: Can obfuscation bootstrapping be accomplished under the iO definition while avoiding an exponential loss in security and without relying on FHE?

Many constructions and reductions related to iO result in an exponential loss in security [15] and even results exist showing certain black-box iO derived reductions require this loss [3]. The open question of an improved obfuscation bootstrapping construction may result in an impossibility result or a reduction to another problem.

1.2 Preliminaries

In this section we explain the definitions and constructions to be used in the proofs. Many of these definitions are taken from [1].

1.2.1 Definitions

In the following definitions, for any set S, $s \overset{\$}{\leftarrow} S$ refers to setting a element s to a random element in the set S. For a randomized algorithm $A(x)$, $A(x;r)$ refers to the output of A on input x using the random coins r. $y \overset{\$}{\leftarrow} A(x)$ refers to setting y to the output of $A(x;r)$ where the random coins, r, are randomly chosen.

Definition 1.2.1. (Circuit Families). Let $\mathcal{F}$ be an infinite sequence of circuit families $\{\mathcal{F}_n\}_{n\in\mathbb{N}}$ where for every $n \in \mathbb{N}$, $\mathcal{F}_n$ is a set of Boolean circuits with n inputs, $m(n)$ outputs, and a circuit size $\ell(n)$ where m and ℓ are bounded by a polynomial of n.

Definition 1.2.2. (Pseudorandom functions). Let $\mathcal{H} = \{\mathcal{H}_n\}_{n\in\mathbb{N}}$ be a family of polynomially-sized Boolean circuits. Let $\mathcal{H}.\mathcal{K}$ be a PPT sampling algorithm that on input 1^n samples a circuit h in $\mathcal{H}_n$. $\mathcal{H}$ is a pseudorandom function family (PRF) if there exists a negligible function $neg(n)$ such that for every non-uniform oracle aided PPT adversary $\mathcal{A}$:

$$\left| \Pr_{h \overset{\$}{\leftarrow} \mathcal{K}(1^n)} [\mathcal{A}^h = 1] - Pr[A^{R_n} = 1] \right| \le neg(n)$$

where R_n is a uniformly random function with the same input and output sizes as $\mathcal{H}_n$.

Definition 1.2.3. (Puncturable pseudorandom functions) Let $\mathcal{H} = \{(\mathcal{H})_n\}_{n\in\mathbb{N}}$ be a family of polynomially-sized Boolean circuits. Let $\mathcal{K}$ be a PPT sampling algorithm that on input 1^n samples a random key k. $\mathcal{H}$ is a puncturable pseudorandom function family if in addition to satisfying the definition of a pseudorandom function using the sampled circuit written as $\mathcal{H}_k$, there exists a polynomial time algorithm *Puncture* which satisfies the following properties:

- **Correctness.** For all keys $k \in \mathcal{K}(1^n)$, all inputs $i \in \{0,1\}^n$, all $x \neq i$, and all punctured keys $k_{-i} \overset{\$}{\leftarrow} Puncture(k,i)$, $\mathcal{H}_{k_{-i}}(x) = \mathcal{H}_k(x)$.

- **Pseudorandom at punctured point.** For every (two-stage) PPT adversary $(\mathcal{A}_1, \mathcal{A}_2)$, where $\mathcal{A}_1$ returns a point $i \in \{0,1\}^n$, and state σ, there is a negligible function $neg(n)$ such that for k randomly sampled from $\mathcal{K}(1^n)$ and $k_{-i} \overset{\$}{\leftarrow} Puncture(k, i)$,

  $$|Pr[\mathcal{A}_2(\sigma, k_{-i}, i, \mathcal{H}_k(i)) = 1] - Pr[\mathcal{A}_2(\sigma, k_{-i}, i, \$) = 1]| \leq neg(n)$$

  where \$ is a uniformly selected random bitstring of size equal to the output size of $\mathcal{H}_k$.

Definition 1.2.4. (Strong pseudorandom permutations). Let $\mathcal{P} = \{\mathcal{P}_n\}_{n\in\mathbb{N}}$ be a family of polynomially-sized Boolean circuits evaluating an n-bit permutation. Let $\mathcal{P}.\mathcal{K}$ be a PPT sampling algorithm that on input 1^n samples a circuit p in $\mathcal{P}_n$. $\mathcal{P}$ is a strong pseudorandom permutation family (sPRP) if there exists a negligible function $neg(n)$ such that for every non-uniform oracle aided PPT adversary $\mathcal{A}$:

$$\left| \underset{p \overset{\$}{\leftarrow} \mathcal{K}(1^n)}{Pr} [\mathcal{A}^{p,p^{-1}} = 1] - Pr[\mathcal{A}^{R_n, R_n^{-1}} = 1] \right| \leq neg(n)$$

where R_n and R_n^{-1} are a uniformly random permutation with the same input and output sizes as $\mathcal{P}_n$ and its inverse. The adversary is given access to both the permutation p and the inverse of the permutation p^{-1}.

Definition 1.2.5. (Randomized Encoding). Let $F_n : X_n \to Y_n$ be an efficiently computable function. Then a randomized encoding scheme RE for F_n is a tuple of PPT algorithms (En, De, Sim) such that:

- $En : X_n \times R_n \to E_n$

- $De : E_n \to Y_n$

- $Sim : E_n \overset{\$}{\to} Y_n$

We assume the values for X_n, Y_n, E_n, R_n are bit strings. For this scheme to be a secure randomized encoding, the tuple of algorithms must satisfy the following properties:

- **Perfect Correctness** For every $n \in \mathbb{N}$, $x \in \{0,1\}^n$, and $r \in R_n$,

$$De(En(x;r)) = F_n(x)$$

- **Computational Privacy** For every non-uniform PPT oracle aided adversary $\mathcal{A}$,

$$\left|Pr[\mathcal{A}^{En(\cdot;\$)} = 1] - Pr[\mathcal{A}^{Sim(F(\cdot))} = 1]\right| \leq neg(n)$$

  where the function En uses fresh randomness $\$ \in R_n$ in each oracle invocation.

Definition 1.2.6. (Virtual Black-Box Obfuscator). Let $\mathcal{F} = \{\mathcal{F}_n\}_{n\in\mathbb{N}}$ be a family of polynomial-sized Boolean circuits. A virtual black-box (VBB) obfuscator $\mathcal{O}$ for the circuit family $\mathcal{F}$ is a PPT algorithm mapping each circuit $f \in \mathcal{F}_n$ to a new circuit $[f]$ with the following properties:

- **Preserve Functionality.** For every $n \in \mathbb{N}$, $f \in \mathcal{F}_n$, and $x \in \{0,1\}^n$,

$$Pr[[f](x) \neq f(x)] \leq neg(n)$$

- **Polynomial Slowdown.** There exists a polynomial p such that for all $n \in \mathbb{N}$ and $f \in \mathcal{F}_n$, the size of the circuit produced by $\mathcal{O}(f)$ is at most $p(|f|)$.

- **Virtual Black-Box.** For every non-uniform PPT adversary $\mathcal{A}$, there exists an oracle aided PPT simulator Sim such that for every $n \in \mathbb{N}$ and $f \in \mathcal{F}_n$,

$$\left|Pr[\mathcal{A}(\mathcal{O}(f)) = 1] - Pr[Sim^f(1^{|f|}, 1^n) = 1]\right| \leq neg(n)$$

Definition 1.2.7. (Indistinguishability Obfuscator). Let $\mathcal{F} = \{\mathcal{F}_n\}_{n\in\mathbb{N}}$ be a family of polynomial-sized Boolean circuits. An indistinguishability obfuscation (iO) obfuscator $\mathcal{O}$ for the circuit family $\mathcal{F}$ is a PPT algorithm mapping each circuit $f \in \mathcal{F}_n$ to a new circuit $[f]$ with the following properties:

- **Preserve Functionality.** For every $n \in \mathbb{N}$, $f \in \mathcal{F}_n$, and $x \in \{0,1\}^n$,

$$Pr[[f](x) \neq f(x)] \leq neg(n)$$

- **Polynomial Slowdown.** There exists a polynomial p such that for all $n \in \mathbb{N}$ and $f \in \mathcal{F}_n$, the size of the circuit produced by $\mathcal{O}(f)$ is at most $p(|f|)$.

- **Indistinguishability.** For every $n \in \mathbb{N}$, for all equally sized and equivalent circuits $f_0, f_1 \in \mathcal{F}_n$, and for every non-uniform PPT adversary $\mathcal{A}$,

$$|Pr[\mathcal{A}(\mathcal{O}(f_0)) = 1] - Pr[\mathcal{A}(\mathcal{O}(f_1)) = 1]| \leq neg(n)$$

1.2.2 Applebaum Construction

The obfuscation bootstrapping construction presented in [1] relies on 3 primitives to construct the new obfuscation scheme. The first component is a randomized encoding scheme RE for the evaluator F of a circuit family $\mathcal{F}$. The next component is a family of pseudorandom functions $\mathcal{H}$ where the output size of $\mathcal{H}_n$ is equal to the size of the randomness input of $RE.En$. The third component is an obfuscator $\mathcal{O}$ for a circuit family $\mathcal{G}$, which will be defined later.

Combining these components in the construction produces an obfuscator $\widehat{\mathcal{O}}$. In the original construction in [1] $\mathcal{O}$ and $\widehat{\mathcal{O}}$ were VBB obfuscators, but throughout this analysis, we will consider them as iO obfuscators.

Explicit Construction

The circuit family $\mathcal{G} = \{\mathcal{G}_n\}$ is the circuit family that will be needed to be obfuscated by $\mathcal{O}$. We define $\mathcal{G}$ as the circuit family where $\mathcal{G}_n$ contains all the circuits of the form,

$$g_{f,h} : x \mapsto RE.En((f, x), h(x)), \quad \forall f \in \mathcal{F}_n, h \in \mathcal{H}_n \tag{1}$$

Then the Applebaum construction, which we will refer to as $AiO_{\mathcal{O},RE,\mathcal{H}}$, follows:

```
$\underline{AiO_{\mathcal{O},RE,\mathcal{H}}(f)}$
$h \overset{\$}{\leftarrow} \mathcal{H}.\mathcal{K}(1^n)$
$[g] \overset{\$}{\leftarrow} \mathcal{O}(g_{f,h})$
$[f] \leftarrow (x \mapsto RE.De([g](x)))$
return $[f]$
```

where $g_{f,h}$ is as defined in (1).

On return, the output of $AiO_{\mathcal{O},RE,\mathcal{H}}(f) = [f]$ is the composition of the circuit $RE.De$ with $[g]$. This is functionally equivalent to the original circuit f. As shown in [1], $AiO_{\mathcal{O},RE,\mathcal{H}}$ is a secure obfuscation bootstrapping scheme under the VBB definition.

1.2.3 Generalized Construction

The original construction is limited by how it defines the obfuscator $\mathcal{O}$ for the circuit family $\mathcal{G}$ used in the construction $AiO_{\mathcal{O},RE,\mathcal{H}}$. $\mathcal{G}$ is defined as a specific circuit family. When considering the VBB definition, this is not an issue; however, under the iO definition, the circuit family is so limited that trivial constructions of $\mathcal{O}$ can be constructed. This leads to the issue explained in Section 2.1.

Because the construction presented by Applebaum is focused only on the VBB definition, when analyzing the application to the iO definition, we consider stronger requirements of the obfuscator $\mathcal{O}$ used. We modify the circuit family $\mathcal{G}$ that the obfuscator needs to be able to obfuscate to prevent this issue. Define $AiO\_NC_{\mathcal{O},RE,\mathcal{H}}$ as $AiO_{\mathcal{O},RE,\mathcal{H}}$

where $\mathcal{O}$ is an indistinguishability obfuscator for the class of circuits $\mathcal{G} = NC^1$.

1.2.4 Fundamental Lemma of Game Playing

Throughout several proofs, we make use of the Fundamental lemma of game playing as formally defined in [5].

In order to prove that two distributions are computationally indistinguishable from each other, the distributions may be identical until a specific event happens. Treating these distributions as games, if two games execute in a functionally equivalent way under the same adversary until some flag **bad** is set, these games are considered *identical-until-bad* games.

G and H are identical-until-bad games if for any adversary $\mathcal{A}$, the executions of G and H under adversary $\mathcal{A}$ are identical until the event occurs setting the flag **bad** to true. Once this occurs, there is no longer a guarantee the games will have equivalent execution under adversary $\mathcal{A}$.

Let G and H be identical-until-bad games. Let $\mathcal{A}$ be an adversary. Then

$$|Pr[G_{\mathcal{A}} = 1] - Pr[H_{\mathcal{A}} = 1]| \leq Pr[G_{\mathcal{A}} \text{ sets bad}]$$

Chapter 2

Constructions for Proofs

2.1 Trivial Counterexample to Construction

The Applebaum construction $AiO_{\mathcal{O},RE,\mathcal{H}}$ is secure under the VBB definition; however, the construction relies on an obfuscator $\mathcal{O}$ which only needs to obfuscate a narrowly defined circuit family $\mathcal{G}$ composed of circuits evaluating the functions

$$x \mapsto RE.En((f,x),h(x)), \quad \forall f \in \mathcal{F}_n, h \in \mathcal{H}_n$$

where $\mathcal{H}$ is the PRF in $AiO_{\mathcal{O},RE,\mathcal{H}}$ and RE is a randomized encoding scheme for an evaluator F which can evaluate the polynomially-sized circuit family $\mathcal{F}$. When considering the case where $AiO_{\mathcal{O},RE,\mathcal{H}}$ is applied to iO instead of VBB obfuscation, the way $\mathcal{G}$ is defined can lead to a construction that breaks the indistinguishability of the scheme. Constructing an iO obfuscator which trivially satisfies the obfuscation of circuit family $\mathcal{G}$ breaks the security of $AiO_{\mathcal{O},RE,\mathcal{H}}$.

Theorem 2.1.1. Under standard assumptions, there exists a secure iO obfuscator for the circuit family $\mathcal{G}$, a PRF, and randomized encoding scheme such that the construction presented by Applebaum fails to produce a secure iO obfuscator.

We prove this by constructing a randomized encoding $\widehat{RE}$, a PRF $\widehat{\mathcal{H}}$, and an obfuscation scheme $\mathcal{O}$ with an adversary which can break the *iO*-security definition for

$AiO_{\mathcal{O},\widehat{RE},\widehat{\mathcal{H}}}$.

2.1.1 Construction of randomized encoding scheme $\widehat{RE}$

The construction is derived from a randomized encoding scheme RE for the function evaluator F of a family of circuits $\mathcal{F}$. The original randomized encoding scheme, RE, can be derived from Yao's garbled circuit scheme as shown in [2]. We use RE to construct a new randomized encoding scheme $\widehat{RE}$ as follows:

- $\underline{\widehat{RE}.En(x;(r_0,r_1,r_2))}$

  return $\big(RE.En(x;r_1),r_0,r_2\oplus x\big)$

- $\underline{\widehat{RE}.De((y_0,y_1,y_2))}$

  return $RE.De(y_0)$

  where y_0 is the bits of input pertaining to $RE.En(x;r_1)$, y_1 being the bits of r_0, and y_2 is the bits of $r_2\oplus x$.

- $\underline{\widehat{RE}.Sim(F(x))}$

  $r_0,r_2\leftarrow \$$

  return $\big(RE.Sim(F(x)),r_0,r_2\big)$

Proof $\widehat{RE}$ is a randomized encoding scheme. To demonstrate $\widehat{RE}$ is a randomized encoding scheme, we demonstrate it holds perfect correctness and computational privacy.

Perfect Correctness For every $n\in\mathbb{N}$, $x\in\{0,1\}^n$, and $r\in R_n$,

$$\widehat{RE}.De(\widehat{RE}.En(x;r))=F(x)$$

This is satisfied from RE being a randomized encoding for the function F:

$\widehat{RE}.De(\widehat{RE}.En(x;(r_0,r_1,r_2)))$

$=\widehat{RE}.De\big((RE.En(x;r_1),r_0,r_2\oplus x)\big)$

$= RE.De(RE.En(x;r_1))$

$= F(x)$

Computational Privacy For every non-uniform PPT oracle aided adversary $\mathcal{A}$,

$$\left|Pr[\mathcal{A}^{\widehat{RE}.En(\cdot;\$)} = 1] - Pr[\mathcal{A}^{\widehat{RE}.Sim(F(\cdot))} = 1]\right| \leq neg(n)$$

where the function $\widehat{RE}.En$ is using fresh randomness \$ in each oracle invocation. By construction, the advantage of adversary $\mathcal{A}$ is equal to:

$$\left|Pr[\mathcal{A}^{\underset{r_0,r_1,r_2\leftarrow\$}{RE.En}(\cdot;r_1),r_0,r_2\oplus x} = 1] - Pr[\mathcal{A}^{\underset{r_0,r_2\leftarrow\$}{RE.Sim}(F(\cdot)),r_0,r_2} = 1]\right|$$

where each \$ is fresh randomness of the corresponding sizes for each query to the oracle. To show there is a negligible advantage for the adversary, first examine that r_0, r_1, and r_2 are independent random values. Then, because r_2 is only used once and as a one-time pad,

$$\left|Pr[\mathcal{A}^{\underset{r_0,r_1,r_2\leftarrow\$}{RE.En}(\cdot;r_1),r_0,r_2\oplus x} = 1] - Pr[\mathcal{A}^{\underset{r_0,r_1,r_2\leftarrow\$}{RE.En}(\cdot;r_1),r_0,r_2} = 1]\right| = 0$$

Then the advantage that an adversary can break computation privacy, using the triangle inequality, is bounded by

$$\begin{aligned}\Bigg|Pr[\mathcal{A}^{\underset{r_0,r_1,r_2\leftarrow\$}{RE.En}(\cdot;r_1),r_0,r_2\oplus x} = 1] - \Big(Pr[\mathcal{A}^{\underset{r_0,r_1,r_2\leftarrow\$}{RE.En}(\cdot;r_1),r_0,r_2} = 1] \\ -Pr[\mathcal{A}^{\underset{r_0,r_1,r_2\leftarrow\$}{RE.En}(\cdot;r_1),r_0,r_2} = 1]\Big) - Pr[\mathcal{A}^{\underset{r_0,r_2\leftarrow\$}{RE.Sim}(F(\cdot)),r_0,r_2} = 1]\Bigg|\end{aligned}$$

$$= \left|Pr[\mathcal{A}^{\underset{r_0,r_1,r_2\leftarrow\$}{RE.En}(\cdot;r_1),r_0,r_2} = 1] - Pr[\mathcal{A}^{\underset{r_0,r_2\leftarrow\$}{RE.Sim}(F(\cdot)),r_0,r_2} = 1]\right|$$

Now the only difference between the distributions is the bits corresponding to $RE.En(\cdot;r_1)$ and $RE.Sim(F(\cdot))$ as the remaining bits are equivalent and independent of the first bits for both distributions. By the definition of the randomized encoding scheme RE, any adversary would have negligible advantage distinguishing between $RE.En(\cdot; r_1 \leftarrow \$)$ and $RE.Sim(F(\cdot))$.

This implies any PPT adversary is bounded in the advantage of distinguishing between the encoding and simulator by a negligible function, satisfying computational privacy. Satisfying the correctness and privacy properties proves that the scheme $\widehat{RE}$ is a secure randomized encoding scheme.

□

2.1.2 Constructing PRF $\widehat{\mathcal{H}}$

For the counter example, we need to use a PRF $\widehat{\mathcal{H}}$ which when used as a circuit family, contains no two circuits which are functionally equivalent. This can be constructed from any PRF $\mathcal{H}$ where the function can be sampled by picking a random key $k \in \{0,1\}^n$, written as $\mathcal{H}_k$ and is of the form $\mathcal{H}_k : \{0,1\}^n \rightarrow \{0,1\}^{3n}$.

The constructed PRF $\widehat{\mathcal{H}}$ is sampled as $\widehat{\mathcal{H}}_k$ where k is uniformly sampled from $\{0,1\}^n$. $\widehat{\mathcal{H}}_k$ is written below.

$$\widehat{\mathcal{H}}_k(x) = \begin{cases} 0^{3n} & \text{if } x = k \\ 1^{3n} & \text{if } \mathcal{H}_k(x)[1:n] = 0^n \\ \mathcal{H}_k(x) & \text{otherwise} \end{cases}$$

where $\mathcal{H}_k(x)[1:n]$ is the first n bits of the output.

Proof $\widehat{\mathcal{H}}$ is a PRF. For any sampled key k, by construction $\widehat{\mathcal{H}}_k$ is equal to $\mathcal{H}_k$ for all inputs except where $x = k$ and $\mathcal{H}_k(x)[1:n] = 0^n$.

Because of this equivalence, the only way a PPT adversary will be able to distinguish between $\widehat{\mathcal{H}}_k$ and $\mathcal{H}_k$ would be to find where $x = k$ or the first n bits of $\mathcal{H}_k(x) = 0^n$. Using the fundamental game playing lemma, the advantage is bounded by the probability that an adversary with oracle access to $\mathcal{H}_k$ can query with input x such that $x = k$ or $\mathcal{H}_k(x)[1 : n] = 0$.

Because $\mathcal{H}$ is a PRF, there is a negligible probability that any adversary can find any input x that will result in the first n bits of $\mathcal{H}_k(x)$ being equal to any n-bit constant. If such an adversary existed, it would break the security of $\mathcal{H}$ as it should be indistinguishable from a random function. A random function has a 2^{-n} chance of having the first n bits of output equal to any constant. A PPT adversary which could find query this input with non-negligible probability would break the PRF security of $\mathcal{H}$.

If an PPT adversary $\mathcal{A}$ with oracle access to $\mathcal{H}_k$ could query with the input $x = k$ with non-negligible probability, $\mathcal{H}$ could be broken in another trivial reduction. An adversary $\mathcal{B}$ can be constructed which simply runs the adversary $\mathcal{A}$ with oracle access to $\mathcal{H}_k$, recording the queries. Then $\mathcal{B}$ simply checks if one of the queried inputs, when treated as a key, is consistent with the outputs of the queried inputs.

The probability of either case is negligible, so there is a negligible advantage in distinguishing between $\widehat{\mathcal{H}}_k$ and $\mathcal{H}_k$. Because $\mathcal{H}$ is a PRF, $\widehat{\mathcal{H}}$ is a PRF as well.

□

Furthermore, representing $\widehat{\mathcal{H}}$ as a keyed circuit family will have no duplicate circuits. For any $k, k' \in \{0, 1\}^n$, if $\forall x.\ \widehat{\mathcal{H}}_k(x) = \widehat{\mathcal{H}}_{k'}(x)$, then $\widehat{\mathcal{H}}_k(k) = \widehat{\mathcal{H}}_{k'}(k) = 0^{3n}$.

The only way the first n bits of $\widehat{\mathcal{H}}_k(x)$ is equal to 0^n is if $x = k$, which implies $\widehat{\mathcal{H}}_{k'}(k) = 0^{3n} \Rightarrow k = k'$.

This proves that $\widehat{\mathcal{H}}$ is a PRF where no two key sampled functions are functionally equivalent, which means the circuit family for $\widehat{\mathcal{H}}$ can be written with each circuit being functionally unique. Additionally, only the first n bits of output are needed to determine if two instances of $\widehat{\mathcal{H}}$ are equivalent.

2.1.3 Trivial Obfuscation Scheme

We construct an indistinguishability obfuscation scheme $\mathcal{O}$ for a specific circuit family $\mathcal{G}$. By limiting the circuit family needed to be obfuscated, even the identity function $\mathcal{O}(g) = g$ is a valid iO obfuscator as long as there are no two functionally equivalent circuits in $\mathcal{G}$.

Using the construction $\widehat{\mathcal{H}}$ defined in the previous subsection, when used as a keyed circuit family, for any keys $k' \neq k'$, $\widehat{\mathcal{H}}_k$ and $\widehat{\mathcal{H}}_{k'}$ are not functionally equivalent.

We use the randomized encoding scheme $\widehat{RE}$ as defined in the previous subsection which is a randomized encoding for the evaluator $F : \mathcal{F} \times X \to Y$ which can evaluate the circuit family $\mathcal{F}$ with an input.

We define the circuit family $\mathcal{G}$ that $\mathcal{O}$ can obfuscate as the set of circuits of the form

$$g_{f,\widehat{\mathcal{H}}_k} : x \mapsto \widehat{RE}.En((f,x), \widehat{\mathcal{H}}_k(x)), \quad \forall f \in \mathcal{F}, \widehat{\mathcal{H}}_k \in \widehat{\mathcal{H}}$$

The obfuscator, $\mathcal{O}$, being the identity function is $\mathcal{O}(g_{f,h}) = g_{f,h}$.

Proof $\mathcal{O}$ is an iO obfuscator. To prove that $\mathcal{O}$ is a valid obfuscator under the iO definition, it must satisfy the properties Preserve Functionality, Polynomial Slowdown, and Indistinguishability according to Definition 1.2.7. Being the identity function, Preserve Functionality and Polynomial Slowdown are clearly satisfied.

The third property, Indistinguishability, requires that for all equally sized and equivalent circuits $g_{f,\widehat{\mathcal{H}}_k}, g_{f',\widehat{\mathcal{H}}_{k'}} \in \mathcal{G}$, and for every non-uniform PPT adversary $\mathcal{A}$,

$$\left| Pr[\mathcal{A}(\mathcal{O}(g_{f,\widehat{\mathcal{H}}_k})) = 1] - Pr[\mathcal{A}(\mathcal{O}(g_{f',\widehat{\mathcal{H}}_{k'}})) = 1] \right| \leq neg(n)$$

To show that the Indistinguishability property is satisfied, we demonstrate there are no two circuits in $\mathcal{G}$ which are functionally equivalent. Without two different but equivalent circuits in $\mathcal{G}$, Indistinguishability is satisfied.

If two circuits $g_{f,\widehat{\mathcal{H}}_k}$ and $g_{f',\widehat{\mathcal{H}}_{k'}}$ are functionally equivalent, we show $(f,k) = (f',k')$. This is shown directly by assuming that for all inputs x,

$$\widehat{RE}.En((f,x), \widehat{\mathcal{H}}_k(x)) = \widehat{RE}.En((f',x), \widehat{\mathcal{H}}_{k'}(x))$$

Refer to the outputs of $\widehat{\mathcal{H}}_k(x) = r_0, r_1, r_2$ and $\widehat{\mathcal{H}}_{k'}(x) = r'_0, r'_1, r'_2$ where each output is a bit string of size n. By construction of $\widehat{RE}$, the previous statement can be rewritten as

$$RE.En((f,x), r_1), r_0, r_2 \oplus (f,x) = RE.En((f',x), r'_1), r'_0, r'_2 \oplus (f',x)$$

where r_0 and r'_0 refer to the first n bits of the output of $\widehat{\mathcal{H}}_k(x)$ and $\widehat{\mathcal{H}}_{k'}(x)$. If $r_0 = r_1$ for all inputs x, then for the input $x = k$, $\widehat{\mathcal{H}}_k(k)[1:n] = \widehat{\mathcal{H}}_{k'}(k)[1:n] = 0^n$. In the construction $\widehat{\mathcal{H}}_k$, the first n bits of output equals 0^n if and only if the input is equal to the key. This implies $k = k'$.

If $k = k'$, $r_2 = r'_2$ for all inputs. This implies if $r_2 \oplus (f,x) = r'_2 \oplus (f',x)$ for all inputs, then $(f,x) = (f',x)$. This concludes that $f, k = f', k'$.

If each circuit in the circuit family $\mathcal{G}$ is functionally unique, no adversary can exist as a counter example, satisfying the Indistinguishability property for $\mathcal{O}$ is a valid iO obfuscator. With all three properties satisfied, $\mathcal{O}$ is an iO obfuscator for the circuit family $\mathcal{G}$. □

2.1.4 Constructing Adversary

As shown above, the secure indistinguishability obfuscator $\mathcal{O}$ can be constructed as the identity function for the circuit family $\mathcal{G}$ defined by

$$g_{f,\widehat{\mathcal{H}}_k} : x \mapsto \widehat{RE}.En((f,x), \widehat{\mathcal{H}}_k(x)), \quad \forall f \in \mathcal{F}, \widehat{\mathcal{H}}_k \in \widehat{\mathcal{H}}$$

where we additionally assume the circuits in $\mathcal{G}$ have a simple structure in that a polynomial time algorithm exists which for all f and k, given the circuit $g_{f,\widehat{\mathcal{H}}_k}$, returns f. Naturally, the identity function when used for obfuscating $\mathcal{G}$ does not hide anything from an adversary. Then Applebaum's construction using the randomized encoding scheme $\widehat{RE}$, iO obfuscator $\mathcal{O}$, and $\widehat{\mathcal{H}}$ would be a valid instantiation. Using these components would construct

```
$\underline{AiO_{\mathcal{O},\widehat{RE},\widehat{\mathcal{H}}}(f)}$
$\widehat{\mathcal{H}}_k \xleftarrow{\$} \widehat{\mathcal{H}}.\mathcal{K}$
$[g] \leftarrow \mathcal{O}(g_{f,\widehat{\mathcal{H}}_k})$
$[f] \leftarrow (x \mapsto RE.De([g](x)))$
return $[f]$
```

We assume $\mathcal{F}$ contains any two different but functionally equivalent circuits $f_0, f_1 \in \mathcal{F}$. To prove $AiO_{\mathcal{O},\widehat{RE},\widehat{\mathcal{H}}}$ is not an iO obfuscator, we show a polynomial time adversary exists which can break the Indistinguishability property of $AiO_{\mathcal{O},\widehat{RE},\widehat{\mathcal{H}}}$.

Under the Indistinguishability property in the definition of iO, the adversary $\mathcal{A}$ takes as input either $AiO_{\mathcal{O},\widehat{RE},\widehat{\mathcal{H}}}(f_0)$ or $AiO_{\mathcal{O},\widehat{RE},\widehat{\mathcal{H}}}(f_1)$ and must determine whether it was given f_0 or f_1. $\mathcal{A}$ executes as follows:

1. $\mathcal{A}$ is given $AiO_{\mathcal{O},\widehat{RE},\widehat{\mathcal{H}}}(f)$ where f is either f_0 or f_1.

   $$AiO_{\mathcal{O},\widehat{RE},\widehat{\mathcal{H}}}(f) = [f] = (x \mapsto RE.De([g](x)))$$

   Because the output is $(x \mapsto RE.De([g](x)))$, which is the composition of functions $RE.De \circ [g]$, $\mathcal{A}$ recovers the circuit for $[g]$. We assume the composition of circuits is done in a manner that is reversible.

2. $[g]$ is the obfuscation of $g_{f,\widehat{\mathcal{H}}_k}$ using the identity function, so $[g] = \mathcal{O}(g_{f,\widehat{\mathcal{H}}_k}) = g_{f,\widehat{\mathcal{H}}_k}$ must be of the form $(x \mapsto \widehat{RE}.En((f, x), \widehat{\mathcal{H}}_k))$ for some k. Because we assumed the circuits in $\mathcal{G}$ allow for recovery of f for any $g_{f,\widehat{\mathcal{H}}_k}$, the adversary parses $[g]$ to

return 1 if $[g]$ is of the form $x \mapsto \widehat{RE}.En((f_1, x), \_)$ and return 0 of $[g]$ is of the form $x \mapsto \widehat{RE}.En((f_0, x), \_)$ where $\_$ is any value.

Under adversary $\mathcal{A}$, f_1 and f_0 will always be correctly identified. This results in

$$\left| Pr[\mathcal{A}(AiO_{\mathcal{O},\widehat{RE},\widehat{\mathcal{H}}}(f_0)) = 1] - Pr[\mathcal{A}(AiO_{\mathcal{O},\widehat{RE},\widehat{\mathcal{H}}}(f_1)) = 1] \right| = 1$$

This fails the Indistinguishability property for the iO security definition, meaning the instantiation of Applebaum's construction is not a valid iO obfuscator, proving Theorem 2.1.1.

2.2 Counterexample to Generalized Construction

The Applebaum construction, $AiO_{\mathcal{O},RE,\mathcal{H}}$ is secure under the VBB definition of obfuscation, and we demonstrate in the previous section a counterexample exists when the construction uses the iO definition instead. This counterexample arises from the narrow definition requirements presented in [1].

We further examine Applebaum's construction by enforcing a stronger requirement on one of the components used. In the original construction, the base obfuscator $\mathcal{O}$ used only requires it to obfuscate a narrowly defined circuit family $\mathcal{G}$, which ultimately allows for counterexample shown in Section 2.1.

Generalizing this construction by requiring $\mathcal{O}$ to obfuscate a larger circuit family such as those in NC^1, prevents this issue. This leads to the construction as shown in Section 1.2.3, $AiO\_NC_{\mathcal{O},RE,\mathcal{H}}$. However, analysis of the generalized construction $AiO\_NC_{\mathcal{O},RE,\mathcal{H}}$ reveals it is insufficient for the purpose of bootstrapping iO. We demonstrate that counterexamples will still exist.

Theorem 2.2.1. Under standard assumptions, there exists a secure PRF and randomized encoding scheme such that for any iO obfuscator for the class of NC^1, when used in the

Applebaum construction fails to produce an iO-secure bootstrapped obfuscator.

The basis of the counterexample is a modified version of a counterexample used in Barak et al. [4]. There are circuit families which are impossible to obfuscate under the VBB definition because the structure of a circuit can reveal information about the function which oracle access cannot. This relates in the iO setting because the Applebaum construction tries to use a PRF essentially as a VBB obfuscated random oracle. There are PRFs which cannot be obfuscated under the VBB definition, and when used as an obfuscated source of randomness for a larger construction, can ultimately reveal the randomness used by the entire construction.

In order to learn this randomness, we construct a PRF $\widehat{\mathcal{H}}$ which is unobfuscatable, revealing information through any circuit which can evaluate an instance of the PRF. This, when combined with a specific randomized encoding scheme $\widehat{RE}$, allows an adversary to break the iO definition of $AiO\_NC_{\mathcal{O},\widehat{RE},\widehat{\mathcal{H}}}$.

What follows is the construction of the PRF $\widehat{\mathcal{H}}$, randomized encoding scheme $\widehat{RE}$, and adversary $\mathcal{A}$ to prove Theorem 2.2.1.

2.2.1 Construction of PRF $\widehat{\mathcal{H}}$

The PRF $\widehat{\mathcal{H}}$ used in proving Theorem 2.2.1 is constructed by taking a PRF, $\mathcal{H}$, a strong PRP, $\mathcal{P}$, and using it to build the PRF $\widehat{\mathcal{H}}$. We denote a sampled function in the PRF $\mathcal{H}$ as $\mathcal{H}_{k_1}$ where k_1 is a randomly sampled key. Similarly, we denote a randomly sampled function in the PRP $\mathcal{P}$ as $\mathcal{P}_{k_2}$ where k_2 is a randomly sampled key. $\mathcal{H}$ and $\mathcal{P}$ are used to construct the PRF $\widehat{\mathcal{H}}$, sampled as $\widehat{\mathcal{H}}_{k_3}$ where k_3 is a randomly sampled key. The size of the keys used in sampling $\mathcal{H}$, $\mathcal{P}$, and $\widehat{\mathcal{H}}$ may be different sizes and are determined by a security parameter.

The dimensions of the functions $\mathcal{H}_{k_1}$, $\mathcal{P}_{k_2}$, and $\widehat{\mathcal{H}}_{k_3}$ will be sized so that:

$\mathcal{H}_{k_1} : X \to \{0,1\}^y$

$\mathcal{P}_{k_2} : \{0,1\}^{y+1} \to \{0,1\}^{y+1}$

$$\widehat{\mathcal{H}}_{k_3} : X \rightarrow \{0,1\}^{5y+2}$$

where we note that these functions have an implicit security parameter and the size of y grows linearly with the security parameter.

The construction of $\widehat{\mathcal{H}}$ is defined below along with the additionally used families of functions C, E, Hom, and B.

```
$\underline{\widehat{\mathcal{H}}_{k=(k',k'',k''',k^{IV},k^{V},k^{VI},\alpha,\beta)}(X)}$
return $\mathcal{H}_{k'}(X)||C_{k'',\alpha,\beta}(X)||E_{k''',k^{IV},\alpha}(X)||Hom_{k''',k^{V}}(X)||B_{k',k''',k^{VI},\beta}(X)$


$\underline{C_{k'',\alpha,\beta}(X = (\alpha', i, ct_0, ct_1, op, ct[\ ]))}$
if $\alpha' = \alpha$
    then $\beta$
else $\mathcal{H}_{k''}(X)$


$\underline{E_{k''',k^{IV},\alpha}(X = (\alpha', i, ct_0, ct_1, op, ct[\ ]))}$
$\mathcal{P}_{k'''}(\alpha_i||\mathcal{H}_{k^{IV}}(X))$


$\underline{Hom_{k''',k^{V}}(X = (\alpha', i, ct_0, ct_1, op, ct[\ ]))}$
$(m_0, \_) \leftarrow \mathcal{P}^{-1}_{k'''}(ct_0)$
$(m_1, \_) \leftarrow \mathcal{P}^{-1}_{k'''}(ct_1)$
$m' \leftarrow m_0\ op\ m_1$
$\mathcal{P}_{k'''}(m'||\mathcal{H}_{k^{V}}(X))$


$\underline{B_{k',k''',k^{VI},\beta}(X = (\alpha', i, ct_0, ct_1, op, ct[\ ]))}$
if $\mathcal{P}^{-1}_{k'''}(ct[j]) = (\beta_j, \_)\forall j$
    then $\mathcal{H}_{k'}(X)$
else $\mathcal{H}_{k^{VI}}(X)$
```

Notation:

- α_i is the ith bit of α
- β_j is the jth bit of β
- $||$ denotes concatenation
- $\_$ denotes an arbitrary value

Proof $\widehat{\mathcal{H}}$ is a PRF. In order to prove that $\widehat{\mathcal{H}}$ is a PRF as defined in Definition 1.2.2, we demonstrate through a series of hybrids to show that for any oracle aided PPT adversary $\mathcal{A}$:

$$\left| \underset{\mathcal{H}_k \overset{\$}{\leftarrow} \mathcal{K}_n}{Pr} [\mathcal{A}^{\widehat{\mathcal{H}}_k} = 1] - Pr[A^{R_n} = 1] \right| \leq neg(n)$$

We show a series of hybrids $H_0, H_1, H_2, H_3, H_4, H_5, H_6$ where each hybrid is negligibly distinguishability from the previous. This results in the first and last hybrids being indistinguishable, satisfying the definition.

Each hybrid will use a modified version of $\widehat{\mathcal{H}}$ which we will denote as $\widehat{\mathcal{H}}^{H_i}$ where H_i is the corresponding hybrid. We list the variations of $\widehat{\mathcal{H}}$ below along with the specific functions they use. When a part of the key is no longer used, we label it with $\_$.

$\underline{\widehat{\mathcal{H}}^{H_0}_{k=(k',k'',k''',k^{IV},k^{V},k^{VI},\alpha,\beta)}(X)}$

return $\mathcal{H}_{k'}(X)||C_{k'',\alpha,\beta}(X)||E_{k''',k^{IV},\alpha}(X)||Hom_{k''',k^{V}}(X)||B_{k',k''',k^{VI},\beta}(X)$

$\underline{\widehat{\mathcal{H}}^{H_1}_{k=(\_,\_,k''',\_,\_,\_,\alpha,\beta)}(X)}$

return $R_0(X)||C'_{\alpha,\beta}(X)||E'_{k''',\alpha}(X)||Hom'_{k'''}(X)||B'_{k''',\beta}(X)$

```
$\underline{\widehat{\mathcal{H}}^{H_2}_{k=(\_,\_,\_,\_,\_,\_,\alpha,\beta)}(X)}$
return $R_0(X)||C'_{\alpha,\beta}(X)||E''_\alpha(X)||Hom''(X)||B''_\beta(X)$


$\underline{\widehat{\mathcal{H}}^{H_3}_{k=((\_,\_,\_,\_,\_,\_,\alpha,\beta)}(X)}$
return $R_0(X)||R_1(X)||E''_\alpha(X)||Hom''(X)||B''_\beta(X)$


$\underline{\widehat{\mathcal{H}}^{H'_3=H_4}_{k=(\_,\_,\_,\_,\_,\_,\alpha,\_)}(X)}$
return $R_0(X)||R_1(X)||E''_\alpha(X)||Hom''(X)||R_4(X)$


$\underline{\widehat{\mathcal{H}}^{H''_3=H_5}_{k=(\_,\_,\_,\_,\_,\_,\_,\_)}(X)}$
return $R_0(X)||R_1(X)||R_P(0||R_E(X))||R_P(0||R_{Hom}(X))||R_4(X)$
$\underline{\widehat{\mathcal{H}}^{H_6}_{k=(\_,\_,\_,\_,\_,\_,\_,\_)}(X)}$
return $R_0(X)||R_1(X)||R_2(X)||R_3(X)||R_4(X)$
```

The functions used above are defined below:

```
$\underline{C'_{\alpha,\beta}(X = (\alpha', i, ct_0, ct_1, op, ct[\,]))}$
if $\alpha' = \alpha$
    then $\beta$
else $R_1(X)$
```

```
$\underline{E'_{k''',\alpha}(X = (\alpha', i, ct_0, ct_1, op, ct[\,]))}$
$\mathcal{P}_{k'''}(\alpha_i || R_E(X))$
```

```
$\underline{E''_{\alpha}(X = (\alpha', i, ct_0, ct_1, op, ct[\,]))}$
$R_P(\alpha_i || R_E(X))$
```

```
$\underline{Hom'_{k'''}(X = (\alpha', i, ct_0, ct_1, op, ct[\,]))}$
$(m_0, \_) \leftarrow \mathcal{P}^{-1}_{k'''}(ct_0)$
$(m_1, \_) \leftarrow \mathcal{P}^{-1}_{k'''}(ct_1)$
$m' \leftarrow m_0 \ op \ m_1$
$\mathcal{P}_{k'''}(m' || R_{Hom}(X))$
```

```
$\underline{Hom''(X = (\alpha', i, ct_0, ct_1, op, ct[\,]))}$
$(m_0, \_) \leftarrow R_P^{-1}(ct_0)$
$(m_1, \_) \leftarrow R_P^{-1}(ct_1)$
$m' \leftarrow m_0 \ op \ m_1$
$R_P(m' || R_{Hom}(X))$
```

```
$\underline{B'_{k''',\beta}(X = (\alpha', i, ct_0, ct_1, op, ct[\,]))}$
if $\mathcal{P}^{-1}_{k'''}(ct[j]) = (\beta_j, \_) \forall j$
    then $R_0(X)$
else $R_4(X)$
```

```
$\underline{B''_{\beta}(X = (\alpha', i, ct_0, ct_1, op, ct[\,]))}$
if $R_P^{-1}(ct[j]) = (\beta_j, \_) \forall j$
    then $R_0(X)$
else $R_4(X)$
```

The following are random functions:

$R_0 : X \rightarrow \{0,1\}^y$

$R_1 : X \rightarrow \{0,1\}^y$

$R_E : X \rightarrow \{0,1\}^y$

$R_{Hom} : X \rightarrow \{0,1\}^y$

$R_4 : X \rightarrow \{0,1\}^y$

Additionally $R_P : \{0,1\}^{y+1} \rightarrow \{0,1\}^{y+1}$ is a random permutation.

Hybrid H_0: For the PPT adversary $\mathcal{A}$, the hybrid samples a random key k in the key space of $\widehat{\mathcal{H}}$ and returns the output of $\mathcal{A}^{\widehat{\mathcal{H}}_k}$.

Hybrid H_1: First we replace each PRF $\mathcal{H}$ used in $\widehat{\mathcal{H}}_k$ with random functions. H_1

proceeds identically to H_0, except we replace the use of $\widehat{\mathcal{H}}_k$ with $\widehat{\mathcal{H}}_k^{H_1}$. This changes the return statement in $\widehat{\mathcal{H}}_k^{H_1}$ from

return $\mathcal{H}_{k'}(X)||C_{k'',\alpha,\beta}(X)||E_{k''',k^{IV},\alpha}(X)||Hom_{k''',k^{V}}(X)||B_{k',k''',k^{VI},\beta}(X)$
to
return $R_0(X)||C'_{\alpha,\beta}(X)||E'_{k''',\alpha}(X)||Hom'_{k'''}(X)||B'_{k''',\beta}(X)$

The change from $\widehat{\mathcal{H}}^{H_0}$ to $\widehat{\mathcal{H}}^{H_1}$ safely replaces instances of $\mathcal{H}$ with a random function of equivalent dimensions for each key. The keys $k', k'', k^{IV}, k^{V}, k^{VI}$ are only ever used as the keys for calls to $\mathcal{H}_{k'}(X)$, $\mathcal{H}_{k''}(X)$, $\mathcal{H}_{k^{IV}}(X)$, $\mathcal{H}_{k^{V}}(X)$, $\mathcal{H}_{k^{VI}}(X)$ respectively. The security advantage between hybrids H_0 and H_1 is negligible by the strength of the PRF $\mathcal{H}$ which could be shown by trivial reductions.

$$|Pr[H_0] - Pr[H_1]| \leq neg_1(n)$$

where neg_1 is a negligible function.

Hybrid H_2: We next replace the use of the strong PRP from $\widehat{\mathcal{H}}_k^{H_2}$ with a random permutation. H_2 proceeds identically to H_1 except replacing the use of $\widehat{\mathcal{H}}_k^{H_1}$ with $\widehat{\mathcal{H}}_k^{H_2}$. This changes the return statement in $\widehat{\mathcal{H}}_k^{H_2}$ from

return $R_0(X)||C'_{\alpha,\beta}(X)||E'_{k''',\alpha}(X)||Hom'_{k'''}(X)||B'_{k''',\beta}(X)$
to
return $R_0(X)||C'_{\alpha,\beta}(X)||E''_{\alpha}(X)||Hom''(X)||B''_{\beta}(X)$

The change from $\widehat{\mathcal{H}}^{H_1}$ to $\widehat{\mathcal{H}}^{H_2}$ safely replaces the strong PRP used with a random permutation of equivalent dimensions. The key k''' is only used as the key to $\mathcal{P}_{k'''}(X)$. The security advantage between hybrids H_1 and H_2 can be shown to be negligible through a trivial reduction.

$$|Pr[H_1] - Pr[H_2]| \leq neg_2(n)$$

where neg_2 is a negligible function.

Hybrid H_3: H_3 proceeds identically to H_2, except we replace the use of $\widehat{\mathcal{H}}_k^{H_2}$ with $\widehat{\mathcal{H}}_k^{H_3}$ by removing the conditional in C'. This changes the return statement in $\widehat{\mathcal{H}}_k^{H_3}$ from

return $R_0(X)||C'_{\alpha,\beta}(X)||E''_{\alpha}(X)||Hom''(X)||B''_{\beta}(X)$

to

return $R_0(X)||R_1(X)||E''_{\alpha}(X)||Hom''(X)||B''_{\beta}(X)$

Showing that hybrid H_3 is indistinguishable from hybrid H_2 requires several steps. The structure of the proof is to show 3 separate identical-until-bad games. We show the probability H_2 and H_3 are distinguishable for any adversary is bounded by the summation of the probabilities of these bad events occurring in their respective games. This summation is shown to be negligible, resulting in H_3 and H_2 being indistinguishable.

The game $\texttt{Bad}_i^H$ records the inputs to the oracle that the adversary for some hybrid H makes, and the game uses the recorded inputs to determine if the associated bad event which would set a flag **bad** to true occurred during the execution. $\texttt{Bad}_i^H$ returns 1 if the bad event occurred and returns 0 otherwise. The bad events of each i in $\texttt{Bad}_i^H$ are defined as follows:

- $\texttt{Bad}_1^H$: the adversary in hybrid H makes the query to $\widehat{\mathcal{H}}_k^H$ with input $X = (\alpha', i, ct_0, ct_1, op, ct[\,]))$ where $\alpha'{=}\alpha$.
- $\texttt{Bad}_2^H$: the adversary in hybrid H makes the query to $\widehat{\mathcal{H}}_k^H$ with input $X = (\alpha', i, ct_0, ct_1, op, ct[\,]))$ where $R_P^{-1}(ct[j]) = (\beta_j, \_)\ \forall j$.

- $\mathtt{Bad}_3^H$: the adversary in hybrid H makes a query to $\widehat{\mathcal{H}}_k^H$ resulting in a collision on the last y bits of input to R_P. This can also be written as the adversary making queries with a collision to R_E, R_{Hom}, or finding inputs X, X' where $R_E(X) = R_{Hom}(X')$

These defined bad events are combined with additionally defined hybrids H_3', and H_3'' to prove a negligible difference between hybrids H_3 and H_2 using the fundamental lemma of game playing [5]. By showing

$$|Pr[H_2] - Pr[H_3]| \leq Pr[\mathtt{Bad}_1^{H_3}]$$

$$\left|Pr[\mathtt{Bad}_1^{H_3}] - Pr[\mathtt{Bad}_1^{H_3'}]\right| \leq Pr[\mathtt{Bad}_2^{H_3'}]$$

$$\left|Pr[\mathtt{Bad}_1^{H_3'}] - Pr[\mathtt{Bad}_1^{H_3''}]\right| \leq Pr[\mathtt{Bad}_3^{H_3''}]$$

The following is true:

$$|Pr[H_2] - Pr[H_3]| \leq Pr[\mathtt{Bad}_1^{H_3''}] + Pr[\mathtt{Bad}_2^{H_3'}] + Pr[\mathtt{Bad}_3^{H_3''}] \leq neg_3(n)$$

This is shown using the following lemmas.

Lemma 2.2.1. The execution of H_2 and H_3 are identical until bad.

$$|Pr[H_2] - Pr[H_3]| \leq Pr[\mathtt{Bad}_1^{H_3}]$$

Proof of Lemma 2.2.1. In both hybrids H_2 and H_3, while each hybrid's adversary, $\mathcal{A}$, does not query $\widehat{\mathcal{H}}_k^{H_2}$ or $\widehat{\mathcal{H}}_k^{H_3}$ with input $X = (\alpha', i, ct_0, ct_1, op, ct[\,]))$ where α'=α, by construction of C', $\widehat{\mathcal{H}}_k^{H_2}$ and $\widehat{\mathcal{H}}_k^{H_3}$ are execute equivalently returning

$$R_0(X)||R_1(X)||E''_\alpha(X)||Hom''(X)||B''_\beta(X)$$

This implies the execution of the hybrids are identical until the bad event in $\mathtt{Bad}_1^{H_2}$ or $\mathtt{Bad}_1^{H_3}$. By the fundamental lemma of game playing this is

$$|Pr[H_2] - Pr[H_3]| \leq Pr[\mathtt{Bad}_1^{H_3}]$$

□

The next lemma uses the hybrid H_3' which is identical to the hybrid H_3 except $\widehat{\mathcal{H}}^{H_3}$ is changed to $\widehat{\mathcal{H}}^{H_3'}$ which removes the conditional in B'' changing the return statement from

return $R_0(X)||R_1(X)||E''_\alpha(X)||Hom''(X)||B''_\beta(X)$

to

return $R_0(X)||R_1(X)||E''_\alpha(X)||Hom''(X)||R_4(X)$

Lemma 2.2.2. The execution of $\mathtt{Bad}_1^{H_3}$ and $\mathtt{Bad}_1^{H_3'}$ are identical until bad.

$$\left|Pr[\mathtt{Bad}_1^{H_3}] - Pr[\mathtt{Bad}_1^{H_3'}]\right| \leq Pr[\mathtt{Bad}_2^{H_3'}]$$

Proof of Lemma 2.2.2. In both $\mathtt{Bad}_1^{H_3}$ and $\mathtt{Bad}_1^{H_3'}$, if the adversary does not make queries to $\widehat{\mathcal{H}}^{H_3}$ or $\widehat{\mathcal{H}}^{H_3'}$ with input $X = (\alpha', i, ct_0, ct_1, op, ct[\,]))$ where $R_P^{-1}(ct[j]) = (\beta_j, \_)\forall j$, the outputs of $\widehat{\mathcal{H}}^{H_3}$ and $\widehat{\mathcal{H}}^{H_3'}$ have equivalent execution, returning

$$R_0(X)||R_1(X)||E''_\alpha(X)||Hom''(X)||R_4(X)$$

This implies the hybrids H_3 and H_3' are equivalent until the event in $\mathtt{Bad}_2$ occurs for $\mathtt{Bad}_1^{H_3}$ or $\mathtt{Bad}_1^{H_3'}$. While the games $\mathtt{Bad}_1^{H_3'}$ and H_3' are different, $\mathtt{Bad}_1^{H_3'}$ executes H_3' to check if the bad event occurs, so the event in $\mathtt{Bad}_2$ occurring in game $\mathtt{Bad}_1^{H_3'}$ implies $\mathtt{Bad}_2^{H_3'}$ under the same adversary. By the fundamental game playing lemma,

$$\left| Pr[\mathtt{Bad}_1^{H_3}] - Pr[\mathtt{Bad}_1^{H_3'}] \right| \leq Pr[\mathtt{Bad}_2 \text{ occurs in game } \mathtt{Bad}_1^{H_3'}] \leq Pr[\mathtt{Bad}_2^{H_3'}]$$

□

$\mathtt{Bad}_2^{H_3'}$ **is negligible:** $Pr[\mathtt{Bad}_2^{H_3'}]$ can be bounded by a negligible function. In order for the bad event in $\mathtt{Bad}_2^{H_3'}$ to occur, the adversary must query $\widehat{\mathcal{H}}^{H_3'}$ with input $X = (\alpha', i, ct_0, ct_1, op, ct[\ ]))$ where $R_P^{-1}(ct[j]) = (\beta_j, \_)\forall j$. β is uniformly random y-bit string and is not used anywhere in $\widehat{\mathcal{H}}^{H_3'}$. There are 2^y equally-likely sequences of $(\beta_1, \_), (\beta_2, \_), ..., (\beta_y, \_)$ and the adversary must produce the sequence. Even an adversary with oracle access to R_P^{-1} is bounded by a $\frac{q}{2^y}$ chance of correctly predicting the sequence after q guesses. This implies a negligible probability for $\mathtt{Bad}_2^{H_3'}$ assuming y grows linearly.

The next lemma uses the hybrid H_3'' which is identical to the hybrid H_3' except $\widehat{\mathcal{H}}^{H_3'}$ is changed to $\widehat{\mathcal{H}}^{H_3''}$ changing the return statement from

return $R_0(X)||R_1(X)||E_\alpha''(X)||Hom''(X)||R_4(X)$

to

return $R_0(X)||R_1(X)||R_P(0||R_E(X))||R_P(0||R_{Hom}(X))||R_4(X)$

Lemma 2.2.3. The execution of $\mathtt{Bad}_1^{H_3'}$ and $\mathtt{Bad}_1^{H_3''}$ are identical until bad.

$$\left| Pr[\mathtt{Bad}_1^{H_3'}] - Pr[\mathtt{Bad}_1^{H_3''}] \right| \leq Pr[\mathtt{Bad}_3^{H_3''}]$$

Proof. $\widehat{\mathcal{H}}^{H_3'}$ and $\widehat{\mathcal{H}}^{H_3''}$ have identical execution except in the functions $E_\alpha''(X)||Hom''(X)$ and $R_P(0||R_E(X))||R_P(0||R_{Hom}(X))$ respectively.

For this proof, consider $\widehat{\mathcal{H}}^{H_3''}$ and $\widehat{\mathcal{H}}^{H_3'}$ as keeping state to generate the random permutation R_P dynamically as a table of input/output. Functionally this is equivalent to the stateless distribution where R_P is randomly chosen on construction.

Because R_P is a random permutation, as long as the input to R_P is never repeated, a fresh element from the remaining domain of R_P is generated and returned. In the difference between $\widehat{\mathcal{H}}^{H_3''}$ and $\widehat{\mathcal{H}}^{H_3'}$, $E_\alpha''(X)||Hom''(X)$ and $R_P(0||R_E(X))||R_P(0||R_{Hom}(X))$ both are of the form:

$$R_P(\_||R_E(X))||R_P(\_||R_{Hom}(X))$$

Which assuming both functions have the same source of randomness when dynamically generating R_P, are equivalent until given an input where the last y bits of input to R_P is repeated. Because R_P is not used outside these functions, it would imply that $\widehat{\mathcal{H}}^{H_3''}$ and $\widehat{\mathcal{H}}^{H_3'}$ will return the same, freshly generated values until given inputs to the PRF resulting in a collision on R_E, R_{Hom}, or between R_E and R_{Hom}. This is the bad event in defined in $\texttt{Bad}_3$.

Assuming the adversary does not make repeated queries to $\widehat{\mathcal{H}}^{H_3''}$ or $\widehat{\mathcal{H}}^{H_3'}$ with inputs resulting in a collision on R_E, R_{Hom}, or between R_E and R_{Hom}, $\widehat{\mathcal{H}}^{H_3''}$ and $\widehat{\mathcal{H}}^{H_3'}$ are equivalent when R is determined dynamically.

This implies the games $\texttt{Bad}_1^{H_3'}$ and $\texttt{Bad}_1^{H_3''}$ are identical until either the event in $\texttt{Bad}_3^{H_3'}$ or $\texttt{Bad}_3^{H_3''}$. While the games $\texttt{Bad}_1^{H_3''}$ and H_3'' are different, $\texttt{Bad}_1^{H_3''}$ executes H_3'' to check if the bad event occurs, so the event in $\texttt{Bad}_3$ occurring in game $\texttt{Bad}_1^{H_3''}$ implies $\texttt{Bad}_3^{H_3''}$. By the fundamental lemma of game playing,

$$\left|Pr[\texttt{Bad}_1^{H_3'}] - Pr[\texttt{Bad}_1^{H_3''}]\right| \leq Pr[\texttt{Bad}_3 \text{ occurs in game } \texttt{Bad}_1^{H_3''}] \leq Pr[\texttt{Bad}_3^{H_3''}]$$

□

$\mathtt{Bad}_3^{H_3''}$ **is negligible:** Even an adversary which has oracle access to R_E and R_{Hom} has a negligible probability of having the event in $\mathtt{Bad}_3^{H_3''}$ occur. By definition, R_E and R_{Hom} are independent random functions so the output of any input is randomly chosen from 2^y possible values. The probability of a collision in R_E, R_{Hom}, or between R_E and R_{Hom} is negligible assuming y grows linearly in the security parameter.

$\mathtt{Bad}_1^{H_3''}$ **is negligible:** As α is never used in hybrid H_3'', an adversary would need to query $\widehat{\mathcal{H}}^{H_3''}$ with input $X = (\alpha', i, ct_0, ct_1, op, ct[\,]))$ where $\alpha = \alpha'$ without any information about the value of α. With α being chosen uniformly at random, having 2^y equally likely values, any q queries to $\widehat{\mathcal{H}}^{H_3''}$ have at most a $\frac{q}{2^y}$ chance of triggering the event in $\mathtt{Bad}_3^{H_3''}$. This gives the adversary a negligible probability of the event occurring assuming y grows linearly.

Combining the lemmas: Using the Triangle Inequality, the hybrids H_2 and H_3 are indistinguishable.

From Lemma 2.2.1,

$$|Pr[H_2] - Pr[H_3]| \leq Pr[\mathtt{Bad}_1^{H_3}]$$

$$= Pr[\mathtt{Bad}_1^{H_3}] - (Pr[\mathtt{Bad}_1^{H_3'}] - Pr[\mathtt{Bad}_1^{H_3'}])$$

$$\leq \left| Pr[\mathtt{Bad}_1^{H_3}] - Pr[\mathtt{Bad}_1^{H_3'}] \right| + Pr[\mathtt{Bad}_1^{H_3'}]$$

Which by Lemma 2.2.2

$$\leq Pr[\mathtt{Bad}_2^{H_3'}] + Pr[\mathtt{Bad}_1^{H_3'}]$$

$$= Pr[\mathtt{Bad}_2^{H_3'}] + Pr[\mathtt{Bad}_1^{H_3'}] - (Pr[\mathtt{Bad}_1^{H_3''}] - Pr[\mathtt{Bad}_1^{H_3''}])$$

$$\leq Pr[\mathtt{Bad}_2^{H_3'}] + \left| Pr[\mathtt{Bad}_1^{H_3'}] - Pr[\mathtt{Bad}_1^{H_3''}] \right| + Pr[\mathtt{Bad}_1^{H_3''}]$$

Which by Lemma 2.2.3

$$\leq Pr[\mathtt{Bad}_2^{H_3'}] + Pr[\mathtt{Bad}_3^{H_3''}] + Pr[\mathtt{Bad}_1^{H_3''}]$$

As $Pr[\mathtt{Bad}_3^{H_3''}]$, $Pr[\mathtt{Bad}_2^{H_3'}]$, and $Pr[\mathtt{Bad}_1^{H_3''}]$ are shown to be negligible, this implies hybrids H_2 and H_3 are indistinguishable as well.

$$|Pr[H_2] - Pr[H_3]| \leq neg_3(n)$$

where neg_3 is a negligible function.

Hybrid H_4: Hybrid H_4 proceeds identically to H_3, except the use of $\widehat{\mathcal{H}}_k^{H_3}$ is replaced with $\widehat{\mathcal{H}}_k^{H_4}$ which changes the return statement in $\widehat{\mathcal{H}}_k^{H_4}$ from

return $R_0(X)||R_1(X)||E''(X)||Hom''(X)||B''_\beta(X)$

to

return $R_0(X)||R_1(X)||E''(X)||Hom''(X)||R_4(X)$

where R_4 is a uniformly random function of input size $|X|$ and outputting a bit string of size y. This hybrid is equivalent to the hybrid H_3' described previously.

As shown in lemma 2.2.2, the functions $\widehat{\mathcal{H}}_k^{H_3}$ and $\widehat{\mathcal{H}}_k^{H_3'=H_4}$ are equivalent until an adversary makes a query to $\widehat{\mathcal{H}}_k^{H_3}$ or $\widehat{\mathcal{H}}_k^{H_4}$ with input $X = (\alpha', i, ct_0, ct_1, op, ct[\ ]))$ where $R_P^{-1}(ct[j]) = (\beta_j, \_)\forall j$. Because the adversary needs to determine this value without any knowledge of β, the probability of an adversary achieving this is bounded by at most $\frac{q}{2^y}$, where q is the number of queries the adversary makes. This is a negligible advantage for any adversary when y grows linearly in the security parameter.

$$|Pr[H_3] - Pr[H_4]| \leq neg_4(n)$$

where neg_4 is a negligible function.

Hybrid H_5: Hybrid H_5 proceeds identically to H_4, except the use of $\widehat{\mathcal{H}}_k^{H_4}$ is replaced with $\widehat{\mathcal{H}}_k^{H_5}$. This hybrid is equivalent to H_3'' which changes the return statement in $\widehat{\mathcal{H}}_k^{H_5}$ from

return $R_0(X)||R_1(X)||E''_\alpha(X)||Hom''(X)||R_4(X)$

to

return $R_0(X)||R_1(X)||R_P(0||R_E(X))||R_P(0||R_{Hom}(X))||R_4(X)$

As shown in lemma 2.2.3, the functions $\widehat{\mathcal{H}}_k^{H_3'=H_4}$ and $\widehat{\mathcal{H}}_k^{H_3''=H_5}$ are equivalent until the event that $\widehat{\mathcal{H}}_k^{H_4}$ or $\widehat{\mathcal{H}}_k^{H_5}$ is queried with inputs resulting in a collision on R_E, R_{Hom}, or between R_E and R_{Hom}.

R_E and R_{Hom} are independently random functions, and this leads to the probability of finding a collision in R_E, R_{Hom}, or between R_E and R_{Hom} negligible for any adversary. This probability is negligible assuming y grows linearly in the security parameter.

$$|Pr[H_4] - Pr[H_5]| \leq neg_5(n)$$

where neg_5 is a negligible function.

Hybrid H_6: Hybrid H_6 proceeds identically to H_5, except the use of $\widehat{\mathcal{H}}_k^{H_5}$ is replaced with $\widehat{\mathcal{H}}_k^{H_6}$. This hybrid changes the return statement in $\widehat{\mathcal{H}}_k^{H_6}$ from

return $R_0(X)||R_1(X)||R_P(0||R_E(X))||R_P(0||R_{Hom}(X))||R_4(X)$

to

return $R_0(X)||R_1(X)||R_2(X)||R_3(X)||R_4(X)$

where R_2 and R_3 are random functions of input size and output size equal to that of R_P. The gap between H_5 and H_6 is negligible by the fact that because R_P is a random permutation, and $R_P(0||R_E(X))||R_P(0||R_{Hom}(X))$ and $R_2(X)||R_3(X)$ will each produce fresh random $(y+1)$-sized bit strings for each input unless a collision is found on R_E, R_{Hom}, R_2, or R_3. Because these are independent random functions, there is a negligible probability of this occurrence for any adversary, assuming y grows linearly.

$$|Pr[H_5] - Pr[H_6]| \leq neg_6(n)$$

where neg_6 is a negligible function.

Combining Hybrids

The hybrid H_0 is using the original PRF $\widehat{\mathcal{H}}$ and hybrid H_6 instead uses $\widehat{\mathcal{H}}_k^{H_6}$ which is a random function. Through the use of the hybrids above, the difference between hybrids H_0 and H_6 is bounded by

$$|Pr[H_0] - Pr[H_6]| \leq neg_1(n) + neg_2(n) + neg_3(n) + neg_4(n) + neg_5(n) + neg_6(n)$$

This shows the gap between hybrids H_0 and H_6 are negligible, that $\widehat{\mathcal{H}}$ is a secure PRF. □

2.2.2 Construction of randomized encoding scheme $\widehat{RE}$

In the Applebaum's construction, the randomized encoding scheme uses a PRF as a source of randomness. The PRF we will use, constructed in subsection 2.2.1, is designed such that when given as a circuit reveals information about the PRF. Because of this, the randomized encoding scheme we construct is designed to leak most of the circuit of the PRF.

The construction is derived from any randomized encoding scheme RE which encodes the evaluator $F : \mathcal{F} \times X \rightarrow Y$ for a polynomially-sized circuit family $\mathcal{F}$. This can be accomplished through the use of Yao's garbled circuits such as presented in [2]. An additional requirement of RE is that for any inputs $x_0 \neq x_1$, with high probability of the random sampled r, $RE.En(x_0; r) \neq RE.En(x_1; r)$. This requirement can be accomplished in a similar manner to construction presented in Section 2.1 with a one-time pad of the inputs concatenated to end of the randomized encoding.

Using RE, we construct a randomized encoding scheme $\widehat{RE}$ for the evaluator F where the algorithms $(\widehat{RE}.En, \widehat{RE}.De, \widehat{RE}.Sim)$ are described as

- $\underline{\widehat{RE}.En(x; (r_0, r_1))}$

  return $\big(RE.En(x, r_0), r_1\big)$

  where r_0 is the first $|\mathcal{H}_k(x)|$ bits of the random string,

  r_1 is the remaining $|\widehat{\mathcal{H}}_k(x)| - |\mathcal{H}_k(x)|$ bits of the random string,

- $\underline{\widehat{RE}.De((y_0, y_1))}$

  return $RE.De(y_0)$

  where y_0 is the first $|RE.En(x; r_0)|$ bits of the encoding,

  y_1 is the remaining $|\widehat{\mathcal{H}}_k(x)| - |\mathcal{H}_k(x)|$ bits of the encoding

- $\underline{\widehat{RE}.Sim(F(x))}$

  return $\big(RE.Sim(F(x)), \$\big)$

  where the output is padded with a random bit string \$ of size $|\widehat{\mathcal{H}}_k(x)| - |\mathcal{H}_k(x)|$

Proof $\widehat{RE}$ is secure randomized encoding scheme. Using Definition 1.2.5, $\widehat{RE}$ must satisfy the properties Perfect Correctness and Computational Privacy.

- **Perfect Correctness** For every $n \in \mathbb{N}$, $x \in \{0,1\}^n$, and $r \in R_n$,

$$\widehat{RE}.De(\widehat{RE}.En(x; r)) = F(x)$$

  This is satisfied as because RE is a randomized encoding for evaluator F:

$$\begin{aligned}&\widehat{RE}.De\big(\widehat{RE}.En(x;(r_0,r_1))\big)\\&=\widehat{RE}.De\big((RE.En(x;r_0),r_1)\big)\\&=RE.De(RE.En(x;r_0))\\&=F(x)\end{aligned}$$

- **Computational Privacy** For every non-uniform PPT oracle aided adversary $\mathcal{A}$,

  $$\Big|Pr[\mathcal{A}^{\widehat{RE}.En(\cdot;\$)}=1]-Pr[\mathcal{A}^{\widehat{RE}.Sim(F(\cdot))}=1]\Big|\leq neg(n)$$

  where the function $\widehat{RE}.En$ is using fresh randomness \$ in each invocation.

  By construction, the advantage of adversary $\mathcal{A}$ is equal to:

  $$\big|Pr[\mathcal{A}^{RE.En(\cdot;\$),\$}=1]-Pr[\mathcal{A}^{RE.Sim(F(\cdot)),\$}=1]\big|$$

  where each \$ is a different fresh randomness of its respective size. The random string concatenated to the end of $RE.En(\cdot;\$)$ and $RE.Sim(F(\cdot))$ are independent and identically distributed random bit strings which can be ignored. Then the advantage of an adversary is bounded by the advantage of any non-uniform PPT oracle aided adversary $\mathcal{B}$,

  $$\big|Pr[\mathcal{B}^{RE.En(\cdot;\$)}=1]-Pr[\mathcal{B}^{RE.Sim(F(\cdot))}=1]\big|$$

  which by definition of RE being a randomized encoding scheme, is negligible. This satisfies the privacy property.

Satisfying the correctness and privacy properties proves that the scheme $\widehat{RE}$ is a secure randomized encoding scheme of the evaluator F. □

2.2.3 Adversary Construction

Using the $\widehat{RE}$ and $\widehat{\mathcal{H}}$ constructions in the Applebaum construction allows for a PPT adversary breaking the iO security in a similar manner to the Barak et al. paper [4]. In this case the circuit family for $\widehat{\mathcal{H}}$ cannot be obfuscated under the VBB definition, because the structure of any equivalent circuit can be used to learn hidden information about $\widehat{\mathcal{H}}$. The randomized encoding scheme $\widehat{RE}$ produces the circuit evaluating $x \mapsto \widehat{RE}.En((f, x); \widehat{\mathcal{H}}_k(x))$ which leaks the unobfuscatable information that breaks the indistinguishability definition.

Using the PRF $\widehat{\mathcal{H}}$ and $\widehat{RE}$, the Applebaum bootstrapping construction is:

```
$\underline{AiO_{\mathcal{O},\widehat{RE},\widehat{\mathcal{H}}}(f)}$
$\widehat{\mathcal{H}}_k \xleftarrow{\$} \widehat{\mathcal{H}}.\mathcal{K}$
$[g] \xleftarrow{\$} \mathcal{O}(x \mapsto \widehat{RE}.En((f, x); \widehat{\mathcal{H}}_k(x)))$
$[f] \leftarrow (x \mapsto RE.Decode([g](x)))$
return $[f]$
```

First, $[f]$ is just a composition of the functions $[g]$ and $RE.Decode$, which any adversary can decompose to recover $[g]$. $[g]$ is an obfuscation of the function $\widehat{RE}.En((f, x); \widehat{\mathcal{H}}_k(x))$, which is equivalent to being given a circuit of the function

$$x \mapsto RE.En((f, x), \mathcal{H}_{k'}(x)), C_{k'',\alpha,\beta}(x)||E_{k''',k^{IV},\alpha}(x)||Hom_{k''',k^V}(x)||B_{k',k''',k^{VI},\beta}(x)$$

where $k = (k', k'', k''', k^{IV}, k^V, k^{VI}, \alpha, \beta)$.

With this circuit, the adversary can use the individual circuits corresponding to the functions C, E, Hom, and B to recover the value $\mathcal{H}_{k'}(x)$ hidden in the function B. By learning the value of $\mathcal{H}_{k'}(x)$ for some x, the adversary knows the randomness used in the randomized encoding. Then the adversary can break the iO security definition. We explain this in detail.

Recovering $\mathcal{H}_{k'}(x)$ from B

Given input $x = (\alpha', i, ct_0, ct_1, op, ct[\ ]))$, $B_{k',k''',k^{VI}}(x)$ returns $\mathcal{H}_{k'}(X)$ if $\mathcal{P}^{-1}_{k'''}(ct[j]) = (\beta_j, \_)\ \forall j$. Since $\mathcal{P}^{-1}_{k'''}$ is used as a block cipher, the adversary uses the circuits of the functions C, E, and Hom to determine an encryption of β_j for all j.

E returns the encryption of each bit of α, C returns β if the input has $\alpha' = \alpha$, and Hom performs homomorphic operations on the encrypted values. The adversary can treat each as an individual circuit by ignoring the outputs that do not correspond to the individual circuits.Using equal-sized and equivalent circuits $f_0, f_1 \in \mathcal{F}$, the adversary makes a series of calls to the obfuscated circuits within $[g]$.

1. The adversary recovers encryptions of the bits of α by making $|\alpha|$ queries to E, changing the value of i to get each bit.

2. The adversary using Hom homomorphically computes the circuit corresponding to C using the encrypted input of $x = (\alpha', i, ct_0, ct_1, op, ct[\,]))$ where the encrypted bits of α' are set to the encrypted bits of α. The remaining input of $x = (\alpha', i, ct_0, ct_1, op, ct[\,]))$ does not matter and can be encryptions of arbitrary bits in α. The output of computing C homomorphically will be encryptions of each of the bits of β.

3. The adversary calls the function B with an input $x = (\alpha', i, ct_0, ct_1, op, ct[\ ]))$ where the list of ciphertexts $ct[\ ]$ is the encryptions of the bits of β. B will then return $\mathcal{H}_{k'}(x)$.

4. The adversary determines if the output of the last query to $[g](x)$ is equal to $RE.En((f_0, x), \mathcal{H}_{k'}(x))$ or $RE.En((f_1, x), \mathcal{H}_{k'}(x))$ and returns a 1 or 0 respectively.

As the adversary will always be able to use $[g]$ to recover $\mathcal{H}_{k'}(x)$ for some x, the first part of $[g](x)$ is guaranteed to be equal to $RE.En((f_0, x), \mathcal{H}_{k'}(x))$ or $RE.En((f_1, x), \mathcal{H}_{k'}(x))$.

The only case where the adversary cannot determine whether f_0 or f_1 is used is when $RE.En((f_0, x), \mathcal{H}_{k'}(x)) = RE.En((f_1, x), \mathcal{H}_{k'}(x))$. We assume RE is chosen such that for any $(f_0, x) \neq (f_1, x)$, that $RE.En((f_0, x), r) \neq RE.En((f_1, x), r)$ with high probability for randomly sampled r.

This implies $RE.En((f_0, x), \mathcal{H}_{k'}(x)) \neq RE.En((f_1, x), \mathcal{H}_{k'}(x))$ with low probability, resulting in $AiO_{\mathcal{O},\widehat{RE},\widehat{\mathcal{H}}}$ being broken, proving Theorem 2.2.1.

Chapter 3

Analysis of Bootstrapping

3.1 Other Forms of Bootstrapping

Because we demonstrate that the construction for bootstrapping obfusction presented in [1] is not secure under the iO definition, the next step is to examine the constructions which do produce valid iO obfuscators. The main two to be examined are the ones by Garg, Gentry, Halevi, Raykova, Sahai, and Waters [14] and by Canetti, Lin, Tessaro, and Vaikuntanathan [1].

For each, we provide the construction, an overview of the proof, an analysis of key aspects of the proofs in each construction, and an overview of potential issues surrounding performance.

3.1.1 Garg et al. Bootstrapping

Shown in [14], Garg et al. present a way to take an obfuscator for a small class of circuits and transform it into an obfuscator for a large class of circuits. The reason for this construction is from the fact that in the same paper, they propose a candidate construction for iO which can only obfuscate circuits within NC^1. To create an obfuscator for general polynomial-sized circuits, they present a way to bootstrap the candidate construction by taking advantage of Fully Homomorphic Encryption (FHE).

In order to obfuscate a circuit C, the idea is to use FHE to homomorphically evaluate the FHE encryption of C as input to a universal circuit. This aspect does not rely on obfuscation and can be performed on polynomial circuits and produces an encrypted output of the function. The only need is to decrypt the output without revealing the circuit or FHE key.

To decrypt the output of the universal circuit, the encrypted output from the universal circuit, along with a proof that the output was correctly computed is given to an obfuscated circuit which checks the proof and decrypts the output if the proof is correct.

The construction makes use of low-depth proofs of FHE computations. In the paper, [14], they describe the way to construct such a proof. Essentially record all the inputs and outputs to each FHE evaluation gate. Then to verify the proof, check that the inputs of each gate evaluate to the specified output, and these outputs match with the inputs to other gates.

This proof technique is low depth because it is only necessary to check each the inputs and outputs of each individual gate, which assuming each FHE operation can be computed as a low depth circuits, can be checked in low depth.

Garg et al. Construction

The obfuscation scheme relies on at a least leveled homomorphic encryption scheme, but we describe it using fully homomorphic encryption. To obfuscate a circuit C, the obfuscation scheme performs the following using the base obfuscator $\mathcal{O}$ and FHE scheme FHE. These have an implicit security parameter λ associated with each function.

1. Generate $(PK_0, SK_0) \overset{\$}{\leftarrow} FHE.KeyGen(\ )$ and $(PK_1, SK_1) \overset{\$}{\leftarrow} FHE.KeyGen(\ )$.

2. Using a Universal Circuit for NC^1 circuits, $U(\cdot,\cdot)$, with the circuit C encoded as input, obtain $g_0 \overset{\$}{\leftarrow} FHE.Enc_{PK_0}(C)$ and $g_1 \overset{\$}{\leftarrow} FHE.Enc_{PK_1}(C)$.

3. Define the function P^{SK_0,g_1,g_2}:

```
$\underline{P^{SK_0,g_1,g_2}(x, e_0, e_1, \phi)}$ :
$t \xleftarrow{\$}$ check $\phi$ is a low-depth proof that:
  $e_0$ and $e_1$ are the FHE evaluations of $U(g_0, x)$ and $U(g_1, x)$
if $t$ then $FHE_{SK_0}.Dec(e_0)$
else 0
```

Using the base obfuscator $\mathcal{O}$, obfuscate a circuit evaluating P^{SK_0,g_1,g_2} which takes as input (x, e_0, e_1, ϕ) where x is the input for the circuit C, e_0 and e_1 are encryptions computed using FHE under PK_0 and PK_1 respectively, and ϕ is a low-depth proof. The obfuscation of P^{SK_0,g_1,g_2} is denoted as $[P]$.

4. Return $[P]$, g_0, g_1, PK_0, and PK_1.

Then $C(x)$ can be evaluated using $[P]$, g_0, g_1, PK_0, and PK_1.

1. Using PK_0 and PK_1, encrypt the input x under each key.
2. Using PK_0,PK_1, g_0, and g_1 along with the encryptions of x to homomorphically evaluate the universal circuit to produce

   $e_0 \leftarrow U(g_0, FHE_{PK_0}.Enc(x))$ and $e_1 \leftarrow U(g_1, FHE_{PK_1}.Enc(x))$. While computing e_0 and e_1, record all inputs and outputs for the homomorphic operations to produce the low-depth proof ϕ.
3. Compute $[P](x, e_0, e_1, \phi)$, which will return the value of $C(x)$.

Overview of Proof

This construction relies on the strength of both the iO obfuscator and FHE with a sequence of 5 hybrids to prove that the obfuscation of two equivalent circuits is computationally indistinguishable. The proof leverages the ability to switch between the two FHE keys used throughout the computation without changing the functionality of the obfuscated circuit $[P]$.

The more detailed proof is found in [14], but we provide an overview of its construction. In order to prove that obfuscations of two functionally equivalent circuits C_0 and C_1 are indistinguishable, the first hybrid is the adversary accessing the default construction using circuit C_0 with the last being the adversary accessing the construction under circuit C_1.

The second hybrid keeps the construction the same except making $g_0 \overset{\$}{\leftarrow} FHE.Enc_{PK_0}(C_0)$ and $g_1 \overset{\$}{\leftarrow} FHE.Enc_{PK_1}(C_1)$. This hybrid is indistinguishable from the previous because they present a reduction which would break the FHE scheme.

$[P]$ in the previous hybrids only used SK_0 to decrypt the encryption e_0. The third hybrid replaces the obfuscated circuit with a functionally equivalent circuit $[P']$ which uses SK_1 to decrypt the encryption e_1 instead of using SK_0 to decrypt e_0. This is secure by the fact that both circuits are functionally equivalent, so by iO, $[P]$ and $[P']$ are indistinguishable.

The fourth hybrid makes $g_0 \overset{\$}{\leftarrow} FHE.Enc_{PK_0}(C_1)$ and $g_1 \overset{\$}{\leftarrow} FHE.Enc_{PK_1}(C_1)$. This hybrid can be argued similarly to the second hybrid where the hybrid is indistinguishable from the previous by the strength of the FHE.

The fifth hybrid goes back to using the original obfuscated circuit $[P]$ instead of $[P']$, which is secure since the functions are equivalent. This is the original obfuscated circuit but using circuit C_1 instead of C_0, satisfying the indistinguishability requirement of the iO definition. The functionality requirement is straightforward, being a decryption of the encryption of the output of the circuit.

Analysis of Construction

This construction, unlike the Canetti et al. construction [1], does not rely in increasing the security parameter to keep security.

While this construction avoids the exponential number of hybrids needed, it still relies on the usage of at least leveled homomorphic encryption. This becomes a non-trivial assumption, because in the desire to bootstrap from some small class of circuits to polynomial circuits, the function P^{SK_0,g_1,g_2} makes use of FHE evaluations for checking that inputs e_0 and e_1 are the FHE evaluations of $U(g_0, x)$ and $U(g_1, x)$ respectively as well as the FHE decryption function to decrypt the final output.

Because P^{SK_0,g_1,g_2} must be in the small class of circuits such as NC^1, this FHE evaluations and decryptions must also be in the same class. While it is not necessarily an unreasonable assumption to think FHE evaluations and decryptions can be done in NC^1, it is still an important assumption to be made.

Additionally when considering performance, the use of FHE may not be ideal. To evaluate an obfuscated circuit, the evaluator will need to essentially perform the FHE computation of a universal circuit at least 4 times; once under each key to get the encrypted output of the universal circuit, then once under each key when the circuit $[P]$ verifies the proof.

Compared to schemes like randomized encodings, FHE can be very slow with some of the best-known constructions taking over 0.1 seconds to evaluate the bootstrapping procedure in FHE [12]. Needing to perform this polynomial number of times could be considerable overhead. There will be additional expected overhead caused by the fact that much of the computation will be in the evaluation of $[P]$. It is unclear what the slowdown will be from $\mathcal{O}$ because the requirement for iO is only a polynomial slowdown. Reducing the size of the circuit evaluating P^{SK_0,g_1,g_2} would be very important for performance.

3.1.2 Canetti et al. Construction

The construction by Canetti et al [11] avoids the need for fully homomorphic encryption, extending the Applebaum construction to produce a secure iO bootstrapping scheme. The construction uses a randomized encoding scheme in a similar manner to the

Applebaum construction to convert a circuit of polynomial depth into a circuit which can be encoded as a circuit in NC^1. Canetti et al.'s extension to the construction solves the problems presented in this paper by modifying the security parameters and making use of puncturable PRFs.

Naturally, these changes make the counterexamples shown previously no longer succeed. The issue that can be analyzed is why these changes are necessary from the Applebaum construction and if there is a way to remove the exponential security loss.

Construction Overview

As the construction is an adaptation to the Applebaum construction, we focus on the main differences, namely the security parameters changes and the use of puncturable PRFs. To examine these, we make explicit the security parameter λ. These constructions rely on sub-exponential secure schemes where a security parameter 1^λ results in an advantage for any PPT adversary of at most $2^{-\lambda^\epsilon}$ where ϵ is a constant in $(0, 1)$ for the specific definition.

The construction still uses a randomized encoding scheme RE for an evaluator F of a circuit family $\mathcal{F}$. We explicitly specify the security parameter used in RE as 1^λ. It is assumed that RE for a security parameter 1^λ has at least sub-exponential security where the distinguishing advantage for any PPT adversary is at most $2^{-\lambda^\epsilon}$ according to Definition 1.2.5.

The Applebaum construction allows for the use of any PRF, whereas this construction is limited to using a puncturable PRF $\mathcal{H}$. For a security parameter 1^λ, the puncturable PRF $\mathcal{H}$ is assumed to have a security advantage bounded by $2^{-\lambda^\epsilon}$ for any PPT adversaries for Definition 1.2.2.

The construction makes use of an iO obfuscator $\mathcal{O}$ which uses a security parameter 1^λ with a similar bounded security advantage of $2^{-\lambda^\epsilon}$ for PPT adversaries for Definition 1.2.7.

The Canetti et al. construction takes RE, $\mathcal{H}$, and $\mathcal{O}$ and picks specific security parameters for each. For RE with a security parameter of 1^λ and given a circuit to obfuscate which is at most λ, define $\lambda' = (\lambda log^2(\lambda))^{1/\epsilon}$. Furthermore, define $\lambda'' \geq \lambda'$.

The Canetti et al. construction is defined

```
$\underline{\widehat{\mathcal{O}}(f)}$
$k \xleftarrow{\$} \mathcal{H}.\mathcal{K}(1^{\lambda'})$
$[g] \xleftarrow{\$} \mathcal{O}(1^{\lambda''}, g_{f,\mathcal{H}_k})$
$[f] \leftarrow (x \mapsto RE.De(1^{\lambda'}, [g](x)))$
return $[f]$


where $g_{f,\mathcal{H}_k}$ is the function $x \mapsto RE.En(1^{\lambda'}, (f, x), \mathcal{H}_k(x))$ and all inputs are padded
to the correct length.
```

Proof Overview

This proof is further detailed in [11]. With many schemes relating to iO, there is an exponential security loss in the proof. This is a result of requiring an exponential number of hybrids to be used in the proof, of which this scheme is no different.

The overall proof involves constructing specific new definitions - probabilistic indistinguishability obfuscation and indistinguishable sampler, which when combined can form an indistinguishability obfuscator. The important step resulting in exponential security loss involves the way the probabilistic indistinguishability obfuscation is constructed and proved.

An overview of the probabilistic indistinguishability obfuscation is to construct an obfuscator which when randomly sampling any two circuits from a specific circuit family, the obfuscations of these circuits are indistinguishable.

Canetti et al. present a way to achieve a probabilistic indistinguishability obfuscation scheme from a one-way function and indistinguishability obfuscator. Randomly sampling the circuits C_0 and C_1 from the distribution, the probabilistic indistinguishability obfuscator can be proved to make them indistinguishable by using an exponential number of hybrids. With the first hybrid being the obfuscation of C_0 and the last being the obfuscation of C_1, the hybrids change one input/output pair of the circuit at a time to gradually shift C_0 to C_1 to show that no PPT adversary can distinguish between the obfuscations.

This results in many intermediate hybrids obfuscating a circuit E_i for $i \in [1...X]$ where $x_1, ..., x_X$ are the canonically ordered elements of the differing domain between C_0 and C_1, where the size of X can be exponential.

$$E_i(x) = \begin{cases} C_1(x; \mathcal{H}_k(x)) & \text{if } x \leq x_i \\ C_0(x; \mathcal{H}_k(x)) & \text{if } x > x_i \end{cases}$$

Because C_0 and C_1 both use a puncturable PRF $\mathcal{H}$, each circuit E_i can be replaced with a functionally equivalent circuit which uses a punctured key k_{-i} on the point x_i. This results in the functionally equivalent circuit E_i':

$$E_i'(x) = \begin{cases} C_1(x; \mathcal{H}_{k_{-i}}(x)) & \text{if } x < x_i \\ y \leftarrow C_1(x; \mathcal{H}_k(x)) & \text{if } x = x_i \\ C_0(x; \mathcal{H}_{k_{-i}}(x)) & \text{if } x > x_i \end{cases}$$

Because the E_i and E_i' are functionally equivalent, their obfuscations will be indistinguishable under iO. Additionally, there is a hybrid for each E_i'':

$$E_i''(x) = \begin{cases} C_1(x; \mathcal{H}_{k_{-i}}(x)) & \text{if } x < x_i \\ y \leftarrow \$ & \text{if } x = x_i \\ C_0(x; \mathcal{H}_{k_{-i}}(x)) & \text{if } x > x_i \end{cases}$$

Then no PPT adversary would be able to distinguishable between the obfuscations between E_i' and E_i'' because both are only different at a single point x_i, which even an adversary knowing the punctured key k_{-i} provides negligible advantage in distinguishing by the strength of the puncturable PRF.

While there are several additional hybrids for each x_i, this technique replaces a single input/output pair of the circuit C_0 to what would be found in C_1. Because there can be an exponential number of input/output pair differences between C_0 and C_1, their proof requires an exponential number of hybrids.

To compensate for the need for exponential hybrids, this scheme increases the security parameters used by the puncturable PRF, randomized encoding scheme, and obfuscator while keeping the input sizes the same.

These puncturable PRFs only puncture a single point, so a natural extension be in puncturing an exponential number of points, such as Boneh and Zhandry show with bit-fixing constraints [8]. Puncturable PRFs that can puncture an exponential number of inputs may not be enough. The hybrids for each E_i'' rely on changing the point at x_i to output a random value, but an exponential number of points cannot be replaced with an exponential number of truly random values while keeping the circuit a polynomial size. This problem may be what is needed to be changed in order to remove the need for an exponential number of hybrids.

Being over to replace an exponential number of input/output pairs at a time for the obfuscation would be a way to remove the exponential loss in security. Then the proof would be nearly identical except use only a polynomial number of hybrids.

Why the Counterexamples no Longer Work

Because this scheme claims to produce a secure iO scheme, the counterexamples previously presented in this paper either no longer apply or are no longer insecure. In analyzing why the Applebaum construction is fundamentally different than the Canetti et al. construction, it is important to note why the counterexamples used in Section 2.1 and

Section 2.2 fail.

The trivial counterexample presented in Section 2.1 is no longer a counterexample due to the way the Canetti et al. specify the base obfuscator $\mathcal{O}$. $\mathcal{O}$ in the Applebaum only needed to obfuscate a very specific circuit family. Specified in [1], this family only includes the circuit family $\mathcal{G} = \{g_{f,h} \mid f \in \mathcal{F}, h \in \mathcal{H}\}$.

The trivial counterexample is considered trivial because it allows the identity function to be a valid iO construction for that specific circuit family, but this is no longer the case when the obfuscator must obfuscate a larger circuit family. This would make the counterexample no longer a valid construction, which is exactly what is done by Canetti et al. They specify that the base obfuscator $\mathcal{O}$ must be able to obfuscate at least the class NC^1. The identity function is not an obfuscator for class NC^1, making this no longer a valid counterexample.

The main counterexample presented in Section 2.2 also no longer works. First is the issue that the counterexample uses a PRF which is not necessarily a puncturable PRF whereas the Canetti et al. construction requires a puncturable PRF. It is unclear if the PRF $\widehat{\mathcal{H}}$ constructed in the counterexample or some variant could be modified into a puncturable PRF while still being used as a counterexample.

Even if the same PRF could be constructed as a puncturable PRF, it would still fail due the relation between the input size and key size. The input to the PRF $\widehat{\mathcal{H}}$ needs to exactly correspond in size to the output in a manner that allows an adversary to recover a secret within any circuit evaluating an instance of the PRF.

While the Canetti et al. construction increases the key size that the PRF uses, the input size intentionally remains the same. The adversary for the counterexample relies on several subsets of the PRF key being equal in size to several subsets of the input to the PRF. When the key grows at a different rate than the input size, with high probability the adversary can no longer manipulate the inputs to recover the hidden information in the PRF. This makes the counterexample no longer work.

Improving the Performance

The Canetti et al. construction relies on an exponential number of hybrids in the proof, resulting in the security loss. To compensate for the security loss, the security parameter is increased from λ to $\lambda' = (\lambda log^2(\lambda))^{1/\epsilon}$.

An area of further research would be to analyze if the Canetti et al. construction can be further modified to avoid needing an exponential number of hybrids, or to show some kind of theoretical need for this security loss.

This construction, even with the increase in parameter, has several advantages to the construction proposed by Garg et al. in [14]. The assumptions are weaker, as the construction relies on a randomized encoding scheme and puncturable PRF, both of which can be derived from one-way functions, in contrast with needing a fully homomorphic encryption scheme that evaluates operations and decrypts in NC^1.

Additionally, needing to increase the security parameter from λ to $\lambda' = (\lambda log^2(\lambda))^{1/\epsilon}$ may not result in significant overhead depending on the base obfuscator $\mathcal{O}$ used in the construction. The offline computation of a garbled circuit is considerably faster than FHE.

Additionally, while examining the security proof of the Canetti et al. construction, we notice the security parameter may not need to be increased to as large. As currently defined, $\lambda' = (\lambda log^2(\lambda))^{1/\epsilon}$. The specific reason for this is to prove that each of the possibly exponential number of hybrids is indistinguishable from the previous by an advantage bounded by $\frac{1}{X2^{log^2\lambda}}$ where X is a value bounded by $2^{-\lambda}$.

Because the puncturable PRF, iO obfuscator, and randomized encoding scheme are each assumed to have a $2^{-\lambda^\epsilon}$ distinguishing gap for some security parameter λ, and $\lambda'' \geq \lambda'$, using these components under either λ' or λ'' when $\lambda' = (\lambda + log^2(\lambda))^{1/\epsilon}$ results in a distinguishability gap of

$$2^{-\lambda'^\epsilon} = 2^{-((\lambda+log^2(\lambda))^{1/\epsilon})^\epsilon} = 2^{-(\lambda+log^2(\lambda))} = \frac{1}{2^\lambda 2^{log^2\lambda}} \leq \frac{1}{X2^{log^2\lambda}}$$

All the hybrids the authors prove rely on this bounded advantage in distinguishing between hybrids, meaning setting $\lambda' = (\lambda log^2(\lambda))^{1/\epsilon}$ is larger parameter than is actually needed. This does not remove the need for exponential hybrids, but it would reduce the security parameter by a possibly nontrivial amount.

3.2 Analysis of Bootstrapping Constructions

Something of interest is the way in which the bootstrapping techniques are proved. In both constructions, the hybrids are able to tradeoff bounding the distinguishability of two consecutive hybrids by switching between the strength of the iO obfuscator and some additional primitive with a publicly revealable element. In the Garg et al. construction, the adversary has access to the public keys to allow both encryption and computation under FHE. With FHE, the construction can switch between hybrids relying on the strength of FHE with the strength of iO.

In the Canetti et al. construction, this is the strength of the punctured PRF, which when given a punctured key, means an adversary can't distinguish between the punctured point and a random value. This implies that in certain hybrids, even if an adversary has access to a punctured key, they still cannot distinguish between the hybrids. This is what allows the proof to replace a single input/output pair, switching between the strength of the iO construction for equivalent circuits, and the strength of the punctured PRF.

Regarding the exponential security loss in the Canetti et al. construction, this is a well-documented issue in constructions related to iO. Many constructions to and from iO constructions incur some form of exponential loss in the security reductions [15], typically by using an exponential number of hybrids replacing one input of a function at a time. This includes building trapdoor permutations [6], constrained PRFs [9], and more. Furthermore, there are even theoretical constraints on constructions from black-box models of iO. There is a guaranteed exponential loss in certain security reductions as shown in [3]. The frequent occurrence of this exponential security loss lead credence to the issue specifically with obfuscation bootstrapping.

Further analysis of the obfuscation bootstrapping techniques is to examine how some separation results or work-arounds for this exponential loss in security associated with the Canetti et al. construction. This leads to a few interesting questions. Can a counterexample to Applebaum's construction exist using a puncturable PRF? Can the exponential security loss be avoided for iO bootstrapping using constructions derived from one-way functions? What is the relationship between obfuscation bootstrapping and the other iO based constructions?

3.3 Conclusion

Indistinguishability obfuscation is a cornerstone of cryptography. From an indistinguishability obfuscation scheme, many other primitives can be achieved. From this is the importance in actually finding practical constructions. One important candidate construction [14] shows promise in the effort to build useful obfuscation. The construction focuses only on obfuscating circuit families within NC^1 as opposed to arbitrary polynomial-sized circuits.

As it is useful obfuscating more than just circuit families in NC^1, to construct the obfuscator for more general circuit families, the authors present a solution to convert the candidate obfuscator for the smaller class of circuits to an obfuscator of the polynomial size circuits.

The three main techniques to accomplish this is to what is proposed in [14], [1], [11]. The first, proposed by Garg et al. obfuscates a circuit C by using fully homomorphic encryption to compute the circuit homomorphically then using an obfuscated circuit which can decrypt the results. While this technique is straightforward, it relies on the use of fully homomorphic encryption achievable in the NC^1, which might be a stronger assumption than is needed. The second solution proposed by Applebaum instead relies on weaker assumptions of one-way functions in NC^1, using only garbled circuits and a PRF. The issue is that the technique is designed for VBB obfuscation and not iO, of which the VBB definition is known to be unachievable. The third solution proposed by Canetti et al. modifies Applebaum's construction to work under the iO definition, but ultimately results in an exponential loss in security.

We examined the constructions proposed by Applebaum and Canetti et al. to examine why the changes made in the latter produces a secure iO bootstrapping technique. It was unknown if the Applebaum construction was sufficient under the iO definition. We determined that the Applebaum construction is insufficient, first by demonstrating a trivial counterexample that takes advantage of the specific way the construction is defined. The original proof in [1] relied on the base obfuscator for a narrowly-defined circuit family, and this was not restrictive enough when applied to the iO setting.

In order to examine the Applebaum construction for iO in a more intended way, we generalized the construction to require the base obfuscator to obfuscate a much larger circuit family. While this prevented the trivial counterexample from existing, ultimately the Applebaum construction was still insufficient. We demonstrate an instantiation of the construction with an adversary to break it by taking advantage of the impossibility results of obfuscating VBB [4].

The Canetti et al. construction is nearly identical to the Applebaum construction, modifying several security parameters and restricting the PRF used to a puncturable PRF. Because the Applebaum construction can be shown to fail in the iO case, it demonstrates that a security gap between Applebaum and Canetti et al.'s constructions exist because of these changes. We further examined both the Garg et al. construction and Canetti et al. construction on why they succeed within their proofs, finding tradeoffs between constructions.

Chapters 1, 2, and 3, in part is currently being prepared for submission for publication of the material. Micciancio, Daniele; Roncevich, Evan. The thesis author was the primary investigator and author of this material. Permission was granted from Daniele Micciancio on this material.

Bibliography

[1] Benny Applebaum. Bootstrapping obfuscators via fast pseudorandom functions. In *ASIACRYPT (2)*, pages 162–172, 2014.

[2] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Computationally private randomizing polynomials and their applications. *Computational Complexity*, 15(2):115–162, 2006.

[3] Gilad Asharov and Gil Segev. Limits on the power of indistinguishability obfuscation and functional encryption. *SIAM Journal on Computing*, 45(6):2117–2176, 2016.

[4] Boaz Barak, Oded Goldreich, Rusell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang. On the (im) possibility of obfuscating programs. In *Annual International Cryptology Conference*, pages 1–18. Springer, 2001.

[5] Mihir Bellare and Phillip Rogaway. Code-based game-playing proofs and the security of triple encryption. In *Advances in Cryptology–EUROCRYPT*, volume 4004, page 10, 2006.

[6] Nir Bitansky, Omer Paneth, and Daniel Wichs. Perfect structure on the edge of chaos. In *Theory of Cryptography Conference*, pages 474–502. Springer, 2016.

[7] Nir Bitansky and Vinod Vaikuntanathan. Indistinguishability obfuscation from functional encryption. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 171–190. IEEE, 2015.

[8] Dan Boneh, Kevin Lewi, and David J. Wu. Constraining pseudorandom functions privately. Cryptology ePrint Archive, Report 2015/1167, 2015. https://eprint.iacr.org/2015/1167.

[9] Dan Boneh and Mark Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. *Algorithmica*, 79(4):1233–1285, 2017.

[10] Ran Canetti and Yilei Chen. Constraint-hiding constrained prfs for nc ˆ 1 from lwe. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 446–476. Springer, 2017.

[11] Ran Canetti, Huijia Lin, Stefano Tessaro, and Vinod Vaikuntanathan. Obfuscation of probabilistic circuits and applications. In *TCC (2)*, pages 468–497, 2015.

[12] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachne. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. Cryptology ePrint Archive, Report 2016/870, 2016. https://eprint.iacr.org/2016/870.

[13] Léo Ducas and Daniele Micciancio. Fhew: bootstrapping homomorphic encryption in less than a second. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 617–640. Springer, 2015.

[14] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. *SIAM Journal on Computing*, 45(3):882–929, 2016.

[15] Sanjam Garg, Omkant Pandey, Akshayaram Srinivasan, and Mark Zhandry. Breaking the sub-exponential barrier in obfustopia. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 156–181. Springer, 2017.

[16] Satoshi Hada. Zero-knowledge and code obfuscation. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 443–457. Springer, 2000.