# Indifferentiability Results and Proofs for Some Popular Cryptographic Constructions

Jaiganesh Balasundaram[*]

July 15, 2014

**Abstract**

The notion of indifferentiability, which is a stronger version of the classic notion of indistinguishability, was introduced by Maurer *et al.* in [MRH03]. Indifferentiability, among other things, gives us a way of "securely replacing" a random oracle of one type by a random oracle of a different type. Most indifferentiability proofs in the literature are very complicated, which makes them difficult to verify and in some cases, has even resulted in them being erroneous [CPS08]. In this paper, we use a simple yet rigorous proof technique for proving indifferentiability theorems. This technique is a generalization of the indistinguishability proof technique used by Bernstein in [Ber05] to prove the security of the Cipher Block Chaining (CBC) construction. We use this technique to prove the indifferentiability result for a very simple construction which processes just two blocks of input. This construction can be viewed as bearing close resemblance to the so called Sponge construction [BDPVA11a], on which the winner of SHA-3 competition [BDPVA11b] is based. Also as a warm up, we prove the indistinguishability result for this construction using the *coupling* argument from probability theory. We also prove the non-indifferentiability result for the CBC construction and some of its standard variants, and survey the indifferentiability and non-indifferentiability results for the Merkle-Damgård (MD) construction, some of its standard variants, and the Feistel construction, from the literature.

## 1 Introduction

A *random oracle* of a certain type, is a large object, chosen uniformly at random from a class of objects of that type. An example of a random oracle is a function $f : \{0,1\}^n \to \{0,1\}^n$ chosen uniformly at random from the set of all functions mapping $\{0,1\}^n$ to $\{0,1\}^n$, and that can be queried as an oracle by an algorithm. Bellare and Rogaway, in [BR93], developed the "random oracle methodology". In this methodology, all the parties, including the adversary have access to a random oracle, and the security of a cryptosystem is proved in this setting. This can be viewed as the proof of security of the cryptosystem in the "ideal world". However, there are no random oracles in the real world, and hence while implementing the cryptosystem in the real world, the random oracle is replaced by an "appropriately chosen" concrete function, like the SHA-2 hash function (they stress that the cryptosystem should be "independent" of the concrete function which replaces the random oracle) and hope that the cryptosystem is secure in the real world. They claim that this method, when properly carried out, leads to secure and efficient protocols. In fact, several widely used cryptosystems that have been proven secure in the random oracle model (for example, the RSAES-OAEP encryption scheme which is a part of the PKCS #1 specification [JK03], is based on the OAEP scheme originally introduced by Bellare *et al.* in [BR95], and versions of which have been proven secure in the random oracle model [Sho01] [FOPS04]), and instantiated with a good hash function in the real world, have resisted attacks in the real world.

[*]University of Toronto. Email: `jaiganesh@cs.toronto.edu`. Supported by MITACS Globalink Graduate Fellowship.

However, Canetti *et al.*, in [CGH04], showed that there exists a cryptosystem (albeit an unnatural one) that is secure in the random oracle model, but has *no* secure implementation in the real world. So, one should be very cautious in using the random oracle model because security of a cryptosystem in the random oracle model does not necessarily imply the security of the cryptosystem in the real world.

However, let us consider a different problem: the problem [1] of "securely replacing" a random oracle of one type by a random oracle of a different type. For example, say the problem of securely replacing a random oracle which takes inputs of arbitrary length by a random oracle which takes inputs of fixed length. By "securely replace", we mean the obvious thing: if a cryptosystem is secure in the random oracle model in which the random oracle is of one type (say arbitrary input length random oracle), then the cryptosystem is secure in the random oracle model in which the random oracle is of another type (say fixed input length random oracle). The notion of *indifferentiability*, introduced by Maurer at al. in [MRH03], gives you a way of rigorously defining this.

Since the notion of indifferentiability is a stronger version of the classic concept of *indistinguishability*, we first give an overview of indistinguishability before giving the high level idea of indifferentiability. We give rigorous definitions for both these notions in the next section.

The concept of indistinguishability was first rigorously defined by Blum and Micali in [BM84]. A cryptographic construction $C$, that takes as input a random member (usually a large object) from a class of objects of a certain type (chosen according to a certain distribution), as an oracle, and outputs an object (that is usually large) of a certain type that can be queried by an algorithm as an oracle (so, effectively it has as input an oracle of a certain type, and outputs an oracle of a certain type), is said to be indistinguishable from ideally distributed version of the object it outputs (which is usually a random member from a class of objects of that type, chosen according to a certain distribution), if no distinguisher, which when given an oracle which is either the object output by the construction $C$ or the ideally distributed version of the object output by $C$, and is allowed to interactively query the oracle, can "significantly" tell the difference. There are different flavours of indistinguishability depending on the running time we allow for the distinguisher and the number of queries we allow it to make to the oracle. The flavour that we will be considering here is what we term "bounded oracle indistinguishability", where we allow the running time of the distinguisher to be unbounded, but it is only allowed to make a bounded number of queries to the oracle. Consider the following popular Cipher Block Chaining (CBC) construction.

**Definition 1.1.** *Let $f$ be a function such that, $f : \{0,1\}^n \to \{0,1\}^n$ for every $n \in \mathbb{Z}^+$. For every $n \in \mathbb{Z}^+$, define $CBC^f : \{0,1\}^{2n} \to \{0,1\}^n$ as follows*

$$CBC^f(xy) = f(f(x) \oplus y)$$

*where $x, y \in \{0,1\}^n$.*

It is a well known result [BKR00] [Ber05] that $CBC^f$, for a uniform random $f$, is indistinguishable from a uniform random function $F$ mapping $\{0,1\}^{2n}$ to $\{0,1\}^n$. However, note that in this model, the distinguisher does not have access to the function $f$. Is this model strong enough for all scenarios? Consider the following real world example. Cryptographic hash constructions like the MD5, SHA1 and SHA2 are usually iterations (using a suitable method) over an underlying compression function that is publicly available. Say that our goal is to prove that a cryptographic hash function, when modelled as iterating over a uniform random compression function (a compression function chosen uniformly at random from the set of all compression functions of its type), appears to be uniformly distributed [2]. In this scenario, it makes sense to allow the

[1] Of course, this problem is only interesting if you have considerable faith in the random oracle model.

[2] One might ask the question "What does this type of modelling accomplish?", as the publicly available compression functions are certainly not uniform random functions. One might say that it says something about the method of iteration, since if we have that a cryptographic hash function, when modelled as iterating over a uniform random compression function, can be significantly told apart from a uniform function having the same domain and range as the hash function, then it means that

adversary to have oracle access to the uniform random compression function, since in the real world, the compression function is publicly available to all the parties, including the adversary. So the indistinguishability model is not a strong enough model for this scenario, since the distinguisher is not allowed to query the uniform random compression function.

The indifferentiability model solves this problem. Let us informally define indifferentiability using the above CBC example. The construction $CBC^f$ (for a uniform random $f$) is said to be *indifferentiable* from a uniform random $F$ mapping $\{0,1\}^{2n}$ to $\{0,1\}^n$, if there exists an "efficient" simulator algorithm $S$, that having oracle access to $F$, tries to simulate $f$ in response to a bounded number of queries made to it, such that, no distinguisher, which when given a pair of oracles, which is either $CBC^f, f$ (for a uniform random $f$), or $F, S^F$ (for a uniform random $F$), and is allowed to interactively query both the oracles, can significantly tell the difference. By an "efficient" simulator, we mean that the simulator has bounded running time (in particular, it makes bounded number of oracle queries to $F$). Again, there are different flavours of indifferentiability depending on the running time we allow for the distinguisher and the number of queries we allow it to make to the oracles. The flavour that we will be considering here is what we term "bounded oracle indifferentiability", where we allow the running time of the distinguisher to be unbounded, but it is only allowed to make a bounded number of queries to the oracles.

This paper primarily deals with indifferentiability results and indifferentiability proofs for some popular cryptographic constructions.

## 1.1 A Note on the Adversaries

Since the adversaries that we consider are distinguishing algorithms that are allowed to have unbounded running time, it sufficient to consider only the deterministic distinguishers. This is because, if there is a probabilistic distinguisher that does well, then we can fix the randomness and get a deterministic distinguisher that does as well as the probabilistic distinguisher.

## 1.2 Motivation for Good Security Proofs

Security proofs form an integral part of cryptography. In addition to being correct and rigorous, it helps if a security proof is easy to understand[3], as it helps the readers to verify the proofs and hence making them less error-prone. As an example, let us consider the different proofs available for the above CBC construction in the literature. Though it was folklore that the construction is secure, the first rigorous treatment was given by Bellare *et al.* in [BKR00]. Though the construction and the security notion (indistinguishability) were quite simple, the proof was enormously complicated. Maurer, in [Mau02], provided a simpler proof of indistinguishability for this construction, using the framework for proving indistinguishability theorems that he introduced in the same paper. Bellare *et al.* further simplified the proof in [BR06] using their so called "Code-based game-playing Framework". In [Ber05], Bernstein gave a very short and elegant, albeit less intuitive proof of security for the same construction. This proof is the simplest rigorous proof of the construction we have seen in the literature so far.

Indifferentiability proofs are infamous for being highly complicated [CPS08][HKT10]. One might argue that, since indifferentiability is a stronger notion than indistinguishability, it is not wrong to expect that the indifferentiability proof of a construction to be at least as complicated as (and in most cases strictly more

there is a problem with the iteration method. However, if we have that the cryptographic hash function (when modelled as iterating over a uniform random compression function) appears to be uniformly distributed, then this doesn't tell us much about the real world version of this hash function, since the real world compression function is certainly not a uniform random function, and can have some bad properties which this model fails to capture. We conclude that we do not have a very good answer for this question, and agree that the motivation here is not very concrete.

[3]Of course, a proof being easy to understand is a very subjective thing, as it depends on many factors like the cryptographic construction under consideration, the security notion we want and even the sophistication of the reader.

complicated than) its corresponding indistinguishability proof. While this is indeed true, the intricacy of many of these proofs have made them difficult to understand and review, and in some cases, resulted in them being erroneous. For example, Coron *et al.* in [CPS08] showed that a 6 round Feistel construction involving six independent uniform random functions mapping $\{0,1\}^n$ to $\{0,1\}^n$ resulted in a permutation (together with its inverse) mapping $\{0,1\}^{2n}$ to $\{0,1\}^{2n}$ that was indifferentiable from a uniform random permutation (together with its inverse) mapping $\{0,1\}^{2n}$ to $\{0,1\}^{2n}$. The proof was quite complicated. However, Holenstein *et al.* in [HKT10] gave a simple distinguishability attack which invalidated the whole proof of [CPS08]. In [HKT10], they prove a similar theorem for a 14 round Feistel construction, which is again quite complicated, but is believed to be true. The point is, an erroneous proof was thought to be correct by (most of) the community for almost 2 years! One wouldn't be entirely wrong if they were to attribute this to the intricacy of the proof that made spotting the error much harder.

This motivates us to ask the following question: Is there a simple yet rigorous proof technique for proving indifferentiability theorems that makes the proofs easy to understand and verify? One of the central goals of this paper is to take steps towards resolving this question.

## 1.3 Our Contribution

As stated above, one of the main contributions of this paper is to use a proof technique for the indifferentiability setting, which builds on the work of Bernstein in [Ber05] and Vaudenay in [Vau03] which were in the indistinguishability setting. The technique we use also bears resemblance to the proofs of Maurer in [Mau02] and Bellare in [BR06] which were again, in the indistinguishability setting. More particularly, consider the following construction which resembles the CBC construction.

**Definition 1.2.** *Let $f$ be a function such that, for every positive even integer $n$, $f : \{0,1\}^n \to \{0,1\}^n$. For every positive even integer $n$, define $I^f : \{0,1\}^n \to \{0,1\}^n$ as follows*

$$I^f(xy) = f(f(x0^{n/2}) \oplus y0^{n/2})$$

*where $x, y \in \{0,1\}^{n/2}$.*

This construction can also be viewed as closely resembling the Sponge Construction [BDPVA11a] on which the recent winner [BDPVA11b] of the SHA-3 competition is based. We describe the proof technique in detail via the proof of the indifferentiability of the construction $I^f$ (for a uniform random $f$), from a uniform random function $F : \{0,1\}^n \to \{0,1\}^n$ (chosen independently of $f$). That is, informally speaking, we give an efficient simulator algorithm $S$ that has oracle access to $F$ and tries to simulate $f$ (in response to a polynomial number of queries made to it), such that no distinguisher that makes a polynomial number of queries to a pair of oracles that is either $(I^f, f)$ or $(F, S^F)$, can distinguish between them, except with negligible probability. Though there are proofs of more complicated versions of this construction (which we will talk about in Section 8) in the literature, to the best of our knowledge, we have not come across a proof of this particular construction, and hence this proof can be considered as one of our contributions in this paper. Also, we find most of these proofs in the literature to be confusing, missing a few vital steps, or a combination of both, and hence they are difficult to understand.

Since we will be closely following Bernstein's technique and adapting it to the indifferentiability setting, we first give an overview of his technique before presenting a brief outline of the technique we use. We present a more detailed outline of the technique we use and an indifferentiability proof of the above construction using this technique in Section 7. Bernstein in [Ber05], shows that $CBC^f$, for a uniform random $f$, is indistinguishable from a uniform random function $F : \{0,1\}^{2n} \to \{0,1\}^n$. The outline is as follows. Let games $G_1$ and $G_2$ correspond to $D$ interacting with distributions $F$ and $CBC^f$ respectively. For each $i \in \{1,2\}$, let $p_i$ be the probability that $D$ accepts in Game $G_i$. We want to show that $|p_2 - p_1|$ is close to 0. Bernstein fixes the query-answer sequence and proves that for each fixed query-answer sequence, the probability (over the choice of $f$) that we get the given answer sequence for the given query sequence in Game $G_2$ is greater than equal to the $(1-\epsilon)$ times the probability (over the choice of $F$) that we get the given answer sequence for the given query sequence in Game $G_1$. Then, he uses this theorem to give a fairly straightforward proof that $|p_2 - p_1| \leq \epsilon$ (Theorem 3.1 in [Ber05]).

We would like to see if we can directly use this technique in proving indifferentiability of the construction $I^f$ (for a uniform random $f$), from a uniform random function $F : \{0,1\}^n \to \{0,1\}^n$. Let $S^F$ be the simulator that we will define in Algorithm 1 of Section 7. Let games $G_1$ and $G_2$ correspond to $D$ interacting with distributions $(F, S^F)$ and $(I^f, f)$ respectively. For each $i \in \{1,2\}$, let $p_i$ be the probability that $D$ accepts in Game $G_i$. We want to show that $|p_2 - p_1|$ is close to 0. Following Bernstein, let us fix the query-answer sequence. If we can prove that for each fixed query-answer sequence, the probability (over the choice of $f$) that we get the given answer sequence for the given query sequence in Game $G_2$ is greater than equal to the $(1-\epsilon)$ times the probability (over the choice of $F$) that we get the given answer sequence for the given query sequence in Game $G_1$, then we can use Theorem 3.1 of [Ber05] to prove that $|p_2 - p_1| \leq \epsilon$ and we will be done. However, in Section 7, we will give a counter-example proving that is indeed not the case. More specifically, we will exhibit a query-answer sequence that happens with a positive probability in Game $G_1$ and zero probability in Game $G_2$. For this query-answer sequence, it is clear that the probability (over the choice of $f$) that we get the given answer sequence for the given query sequence in Game $G_2$ is less than $(1-\epsilon)$ times the probability (over the choice of $F$) that we get the given answer sequence for the given query sequence in Game $G_1$

So, we will rule out such query-answer sequences (that happen with a positive probability in Game $G_1$ and zero probability in Game $G_2$), and prove that for every $D$, probability that $D$ causes a ruled out query-answer sequence in Game $G_1$ is small, say $\epsilon_1$ (where $\epsilon_1$ is a function of the number of queries we allow $D$ to make). Now, for each query-answer sequence that have not been ruled out, we will prove that the probability (over the choice of $f$) that we get the given answer sequence for the given query sequence in Game $G_2$ is greater than equal to the $(1-\epsilon)$ times the probability (over the choice of $F$) that we get the given answer sequence for the given query sequence in Game $G_1$. Using these two results, we will extend Theorem 3.1 of [Ber05] to prove that $|p_2 - p_1| \leq \epsilon + \epsilon_1$. Summarizing the approach,

1. Define a notion of "ruled out query-answer sequences". This notion will include query-answer sequences that occur with zero probability in Game $G_2$ but with some positive probability in Game $G_1$, but also some other query-answer sequences (that we include for convenience, see Section 7 for more details).

2. For every $D$, prove that the probability (over the choice of $F$) that $D$ causes a ruled out query-answer sequence to occur in Game $G_1$ is small, say $\epsilon_1$.

3. Prove that for all fixed query-answer sequences that have not been ruled out, the probability (over the choice of $f$) that we get the given answer sequence for the given query sequence in Game 2 is greater than or equal to $(1-\epsilon)$ times the probability (over the choice of $F$) that we get the given answer sequence for the given query sequence in Game 1.

4. Use (2) and (3) to prove that $|p_2 - p_1| \leq \epsilon + \epsilon_1$.

So, Steps 1 and 2 are a new addition compared to the original proof of [Ber05]. Step 4 is an extension of Theorem 3.1 of [Ber05]. Also, for proving Step 3, we will be using a more intuitive Bayesian approach, as opposed to the counting argument of [Ber05].

Of course, proving the indifferentiability of $I^f$ (for a uniform random $f$) is just a starting step, since the construction processes only two blocks of input. The fact that the output length of the construction equals the input length of the construction makes this indifferentiability proof more of a curiosity than anything else, since the construction by itself doesn't seem very useful in practice. We aim to extend this technique to get the indifferentiability proofs of more general constructions (such as the one similar to the construction $I^f$, but processing inputs of arbitrary length), in our future work. More particularly, consider the following generalization of $I^f$.

**Definition 1.3.** *Let $f$ be a function such that, for every positive even integer $n$, $f : \{0,1\}^n \to \{0,1\}^n$. For every positive even integer $n$, define $I'^f : (\{0,1\}^{n/2})^* \to \{0,1\}^n$ recursively as follows.*

$$I'^f(\epsilon) = 0^n$$

$$I'^f(xy) = f(I'^f(x) \oplus y0^{n/2})$$

*where $x \in (\{0,1\}^{n/2})^*$, $y \in \{0,1\}^{n/2}$ and $\epsilon$ is the empty string.*

We would like to use the above proof technique to prove that $I'^f$ (for a uniform random $f$) is indifferentiable from a uniform random function $F : (\{0,1\}^{n/2})^* \to \{0,1\}^n$. As we had stated earlier, though this construction resembles the sponge construction on which much work has been done, to the best of our knowledge we have not yet found an indifferentiability proof for this particular construction (or even for the simpler case $I^f$ which processes only two blocks of input). People have proved indifferentiability results for much more complicated versions of this construction, like the construction which is similar to $I'^f$, but some bits are truncated from the output [BDPVA08]. People have also given the proof for a technically stronger indifferentiability theorem for this version, where the uniform random function is replaced by a uniform random permutation together with its inverse (technically stronger in the sense that, the simulator in that proof simulates both the permutation and its inverse) [BDPVA08][AMP12]. We briefly talk about the indifferentiability results of constructions like these, in Section 8. Specifically, we prove that the construction $I^p$, for a uniform random permutation $p$ (together with its inverse), is not indifferentiable from a uniform random function $F : \{0,1\}^n \to \{0,1\}^n$, and one of the ways of making the construction work (that is, slightly modifying the construction to getting an indifferentiability theorem out of it) is to drop some bits from the output of $I^p$.

Also, as a warm up before presenting the indifferentiability proof of $I^f$ using this technique, we present the proof of indistinguishabulity of $I^f$ (for a uniform random $f$) from a uniform random function $F : \{0,1\}^n \to \{0,1\}^n$. Though this construction is similar to the CBC construction, it turns out that we can get an indistinguishability proof that is much simpler and more intuitive than the technique Bernstein uses in [Ber05] for the indistinguishability proof of the CBC construction. The trick we use here is called *coupling* [Lin92] [Tho00], from probability theory. The high level idea is as follows. Using the conventions we used before, let games $G_1$ and $G_2$ correspond to $D$ interacting with distributions $F$ and $I^f$ respectively. For each $i \in \{1,2\}$, let $p_i$ be the probability that $D$ accepts in Game $G_i$. We want to show that $|p_2 - p_1|$ is close to 0. That is, we would like to show that with high probability, the outcomes (that is, the answers to the queries that $D$ makes) of both the games are identically distributed. To argue this, we *couple* the games by running both the games together with *common randomness* and argue that with high probability, the outcome of both the games (when run with common randomness) are identical.

We also provide negative results regarding the indifferentiability of the Cipher Block Chaining construction $CBC^f$ (for a uniform random $f$) and some standard variants of the CBC construction.

In the beginning of the paper, we also survey some results regarding the indifferentiability of the Merkle-Damgard (MD) construction and the Feistel construction.

After completing most of this work, we learned that a proof technique similar to the one we use in this paper, had previously been used in the indifferentiability proof of a variant of the MD construction in [CN08a].

## 1.4 Related Work

As we mentioned earlier, the technique we use for proving indifferenitability theorems is an extension of the technique used by Bernstein in [Ber05] to prove the security of the CBC construction in the indistinguishability setting. The technique we use also bears resemblance to the technique used by Vaudney in [Vau03], Maurer in [Mau02] and Bellare *et al.* in [BR06], all in the indistinguishability setting. The construction $I^f$ closely resembles the sponge construction, which was introduced by Bertoni *et al.* in [BDPVA11a]. The indifferentiability proofs for the different versions of sponge construction appear in [BDPVA08], by the same authors. Andreeva *et al.* generalized the sponge construction and proved the indifferentiability results for the same in [AMP12]. The indifferentiability results for the MD construction and some of its standard variants that we survey in this paper, are from [CDMP05]. The indifferentiability results for the Feistel construction that we survey in this paper are from [CPS08],[HKT10] and [MPS12]. The instances of the use of coupling arguments in cryptography that we could find were, for the analysis of RC4 cipher by Mironov in [Mir02], and for the analysis of Feistel networks by Morris *et al.* in [MRS09] and Hoang *et al.* in [HR10].

### 1.5 Organization of the Paper

The paper is organized as follows. In Section 2, we present some notations and definitions that we will be using in the rest of the paper. In Section 3, we present some negative results regarding the indifferentiability of CBC construction and some of its standard variants. In Section 4, we survey some results from the literature regarding the indifferentiability of the MD construction and some of its standard variants. In Section 5, we survey some results from the literature regarding the indifferentiability of the Feistel construction. In Section 6, we present the proof of indistinguishability of the construction $I^f$ (for a uniform random $f$), from a uniform random function $F : \{0,1\}^n \to \{0,1\}^n$ (chosen independently of $f$), using coupling. Our main result is in Section 7, where we give a more detailed outline of the technique we use for proving indifferentiability theorems and present the proof of indifferentiability of the construction $I^f$ (for a uniform random $f$), from a uniform random function $F : \{0,1\}^n \to \{0,1\}^n$ using the technique. In Section 8, we investigate the indifferentiability of construction $I^p$, for a uniform random permutation $p$ (together with its inverse), from a uniform random function $F : \{0,1\}^n \to \{0,1\}^n$. We conclude and present further directions for research in Section 9.

## 2 Preliminaries

### 2.1 Notations

- $\mathbb{N}$ denotes the set of all natural numbers and $\mathbb{Z}^+$ denotes the set of all positive integers.
- For every $m, n \in \mathbb{Z}^+$, $\mathbb{F}_m^n$ denotes the set of all functions mapping $\{0,1\}^n$ to $\{0,1\}^m$
- For every $n \in \mathbb{Z}^+$, $\mathbb{P}_n$ denotes the set of all permutations (together with their inverses) mapping $\{0,1\}^n$ to $\{0,1\}^n$.
- For any distribution $D$, the notation $d \leftarrow D$ denotes randomly choosing a member $d$ according to the distribution $D$.

### 2.2 Indistinguishability

In this subsection, we give a meta-definition of indistinguishability. Since this is just a meta-definition, we will be intentionally vague about the types of some objects here. We will be more explicit as to what these types are, when we use this definition to state indistinguishability theorems about constructions, in the upcoming sections.

Let $\mathcal{F} = \{F_n\}_{n \in \mathbb{Z}^+}$ and $\mathcal{G} = \{G_n\}_{n \in \mathbb{Z}^+}$ be two families of distributions parametrized by $n$. In each family, for each $n$, the distribution is over functions or a tuple of functions having as domain and range a set of strings (the length of which depends on $n$). Let $C$ be a deterministic construction that has as input, the parameter $n$ and a random member $f$ from $F_n$ as an oracle, and outputs $C_n^f \in G_n$ that can be queried by an algorithm. In effect, the construction $C$ can be seen as outputting an oracle. We define what it means for the construction $C$ with oracle access to the distribution family $\mathcal{F}$ to be indistinguishable from the distribution family $\mathcal{G}$.

**Definition 2.1** (Indistinguishability)**.** *A deterministic construction $C$ with oracle access to a distribution family $\mathcal{F}$ is said to be indistinguishable from a distribution family $\mathcal{G}$ if $\{C_n^f \mid f \leftarrow \mathcal{F}\}$ is bounded oracle indistinguishable from $\{g \mid g \leftarrow \mathcal{G}\}$.*

By "bounded oracle indistinguishable" we mean that, for every bounded oracle distinguisher $D$ (we will shortly say what we mean by a bounded oracle distinguisher), the following holds. $D$ is given $1^n$ and an oracle of type $\mathcal{G}$ with parameter $n$, which is either $C_n^f$ (for a random member $f$ from $F_n$) or a random member $g$ from $G_n$. Let $Pr[D^{C_n^f}(1^n) = 1]$ denote the probability (over the choice of $f$) that $D$ accepts if the oracle was $C_n^f$ and $Pr[D^g(1^n) = 1]$ denote the probability (over the choice of $g$) that $D$ accepts if the oracle was $g$. Then, for every $c$ and sufficiently large $n$, we have that

$$|Pr\ [D^{C_n^f}(1^n) = 1] - Pr[D^g(1^n) = 1]| \leq \frac{1}{n^c}$$

By bounded oracle distinguisher, we mean that we do not care about the running time of the distinguisher and we only care that it makes a bounded number of queries to the oracle, that is, the total number of queries made by the distinguisher to the oracle is a polynomial in $n$.

We present an example here. Recall the CBC construction which we defined in the previous section. Using the above meta-definition, we will be stating the theorem that $CBC^f$ (for a uniform random $f : \{0,1\}^n \rightarrow \{0,1\}^n$) is indistinguishable from a uniform random $F : \{0,1\}^{2n} \rightarrow \{0,1\}^n$. We won't be proving this theorem since it is a well known result, as discussed in the previous section.

**Theorem 2.2.** *Let $X_n$ be the uniform distribution on the set $\mathbb{F}_n^{2n}$ and let $Y_n$ be the uniform distribution on the set $\mathbb{F}_n^n$, for every positive integer $n$. Let $\mathcal{X} = \{X_n\}$ and $\mathcal{Y} = \{Y_n\}$. Then, the construction CBC having oracle access to the distribution family $\mathcal{Y}$ is indistinguishable from the distribution family $\mathcal{X}$.*

## 2.3 Indifferentiability

In this subsection, we give a meta-definition of indifferentiability. Since this is just a meta-definition, we will be intentionally vague about the types of some objects here. We will be more explicit as to what these types are, when we use this definition to state indifferentiability theorems about constructions, in the upcoming sections.

Let $\mathcal{F} = \{F_n\}_{n\in\mathbb{Z}^+}$ and $\mathcal{G} = \{G_n\}_{n\in\mathbb{Z}^+}$ be two families of distributions parametrized by $n$. In each family, for each $n$, the distribution is over functions or a tuple of functions having as domain and range a set of strings (the length of which depends on $n$). Let $C$ be a deterministic construction that has as input the parameter $n$ and a random member $f$ from $F_n$ as an oracle, and outputs an oracle $C_n^f \in G_n$ that can be queried by an algorithm. A "simulator" is a probabilistic algorithm that has as input the parameter $n$ and a random member $g$ from $G_n$ as an oracle, and outputs an oracle $S_n^g \in F_n$ that can be queried by an algorithm. We define what it means for the construction $C$ with oracle access to the distribution family $\mathcal{F}$ to be indifferentiable from the distribution family $\mathcal{G}$.

**Definition 2.3** (Indifferentiability)**.** *A deterministic construction $C$ with oracle access to a distribution family $\mathcal{F}$ is said to be indifferentiable from a distribution family $\mathcal{G}$ if there exists a probabilistic, efficient, oracle simualtor algorithm $S$ such that, $\{(C_n^f, f) \mid f \leftarrow \mathcal{F}\}$ is bounded oracle indistinguishable from $\{(g, S_n^g) \mid g \leftarrow \mathcal{G}\}$.*

By "bounded oracle indistinguishable" we mean that, for every bounded oracle distinguisher $D$, the following holds. $D$ is given $1^n$, an oracle of type $\mathcal{G}$ with parameter $n$ (let's call it the Left oracle $L$) and an oracle of type $\mathcal{F}$ with parameter $n$ (let's call it the Right oracle $R$), where the pair of oracles $L, R$, is either $C_n^f, f$ (for a random member $f$ from $F_n$) or $g, S_n^g$ (for a random member $g$ from $G_n$). Let $Pr[D^{C_n^f,f}(1^n) = 1]$ denote the probability (over the choice of $f$) that $D$ accepts if the oracles were $C_n^f, f$ and $Pr[D^{g,S_n^g}(1^n) = 1]$ denote the probability (over the choice of $g$ and the random choices made by the simulator) that $D$ accepts if the oracles were $g, S_n^g$. Then, for every $c$ and sufficiently large $n$, we have that

$$|Pr\ [D^{C_n^f, f}(1^n) = 1] - Pr[D^{g, S_n^g}(1^n) = 1]| \leq \frac{1}{n^c}$$

By bounded oracle distinguisher, we mean that we do not care about the running time of the distinguisher and we only care that it makes a bounded number of queries to the oracles, that is, the total number of queries made by the distinguisher to the oracles is a polynomial in $n$.

We used the term "efficient" while describing the simulator $S$. By this we mean that the simulator $S$ runs in time polynomial in $n$ and the number of queries made to it (and in particular, makes a polynomial number of queries to $g$).

We present an example here, again using the CBC construction. Using the above meta-definition, we will be stating the theorem that $CBC^f$ (for a uniform random $f$) is *not* indifferentiable from a uniform random $F : \{0,1\}^{2n} \to \{0,1\}^n$. We will be proving this in Section 3, by defining a distinguisher $D$ which distinguishes $(CBC^f, f)$ and $(F, S^F)$ with overwhelming probability for any efficient simulator $S$.

**Theorem 2.4.** *Let $X_n$ be the uniform distribution on the set $\mathbb{F}_n^{2n}$ and let $Y_n$ be the uniform distribution on the set $\mathbb{F}_n^n$, for every $n \in \mathbb{Z}^+$. Let $\mathcal{X} = \{X_n\}$ and $\mathcal{Y} = \{Y_n\}$. Then, the construction CBC having oracle access to the distribution family $\mathcal{Y}$ is not indifferentiable from the distribution family $\mathcal{X}$.*

**Remark 2.5.** *Note that this proof of "non-indifferentiability" will be stronger than what Definition 2.3 requires. That is, the definition says, to prove non-indifferentiability it is sufficient to show for any efficient simulator there exists a distinguisher, whereas we will be showing there exists a distinguisher for any efficient simulator.*

## 2.4 Sequential Indifferentiability

Sequential indifferentiability is a weaker form of indifferentiability, introduced by Mandal *et al.* in [MPS12]. In sequential indifferentiability, we consider a restricted class of distinguishers, called the sequential distinguishers, which can only make queries (to the oracles it is provided with) in a specific order. More particularly, a sequential distinguisher $D$ can first query the Right oracle as it wants and then query the Left oracle as it wants. But once it starts querying the Left oracle, it cannot query the Right oracle again. So the definition for sequential indifferentiability is similar to the definition of indifferentiability provided in the previous subsection, with this additional restriction.

It is easy to see that indifferentiability implies sequential indifferentiability, but the other direction is not necessarily true (as proved in [MPS12]).

We present an example here, again using the CBC construction. We will be stating the theorem that $CBC^f$ (for a uniform random $f$) is *not* sequentially indifferentiable from a uniform random $F : \{0,1\}^{2n} \to \{0,1\}^n$. We won't be proving this explicitly but it will be easy to see that the distinguisher we provide for the proof of non-indifferentiability of CBC in Section 3, is actually a sequential distintinguisher.

**Theorem 2.6.** *Let $X_n$ be the uniform distribution on the set $\mathbb{F}_n^{2n}$ and let $Y_n$ be the uniform distribution on the set $\mathbb{F}_n^n$, for every $n \in \mathbb{Z}^+$. Let $\mathcal{X} = \{X_n\}$ and $\mathcal{Y} = \{Y_n\}$. Then, the construction CBC having oracle access to the distribution family $\mathcal{Y}$ is not sequentially indifferentiable from the distribution family $\mathcal{X}$.*

## 3 Indifferentiability Results for the CBC Construction and Some of its Standard Variants

In this section, we present some non-indifferentiability results for the well known Cipher Block Chaining (CBC) construction and some of its standard variants.

### 3.1 CBC construction processing 2 blocks of input

We first restate the definition of the CBC construction processing two blocks of input.

**Definition 3.1.** *Let $f$ be a function such that, $f : \{0,1\}^n \to \{0,1\}^n$ for every $n \in \mathbb{Z}^+$. For every $n \in \mathbb{Z}^+$, define $CBC^f : \{0,1\}^{2n} \to \{0,1\}^n$ as follows*

$$CBC^f(xy) = f(f(x) \oplus y)$$

*where $x, y \in \{0,1\}^n$.*

We will prove that $CBC^f$ (for a uniform random $f$) does not give you indifferentiability from a uniform random function $F$ mapping $\{0,1\}^{2n}$ to $\{0,1\}^n$, by giving a very simple attack. We restate the theorem from the previous section.

**Theorem 3.2.** *Let $X_n$ be the uniform distribution on the set $\mathbb{F}_n^{2n}$ and let $Y_n$ be the uniform distribution on the set $\mathbb{F}_n^n$, for every positive integer $n$. Let $\mathcal{X} = \{X_n\}$ and $\mathcal{Y} = \{Y_n\}$. Then, the construction CBC having oracle access to the distribution family $\mathcal{Y}$ is not indifferentiable from the distribution family $\mathcal{X}$.*

*Proof.* Fix $n$. Here we define a distinguisher $D$ which distinguishes $(CBC^f, f)$ and $(F, S^F)$ with overwhelming probability for any efficient simulator $S$, where $f$ is a uniform random function from $\{0,1\}^n$ to $\{0,1\}^n$ and $F$ is a uniform random function from $\{0,1\}^{2n}$ to $\{0,1\}^n$. $D$ has access to oracles $X : \{0,1\}^{2n} \to \{0,1\}^n$ and $Y : \{0,1\}^n \to \{0,1\}^n$, which are either $(CBC^f, f)$ or $(F, S^F)$. $D$ works as follows,

1. Choose arbitrary $n$-bit strings $a, b$ such that $a \neq b$.
2. Let $\alpha \leftarrow Y(a)$ and $\beta \leftarrow Y(b)$.
3. Let $\gamma \leftarrow X(a\alpha)$ and $\delta \leftarrow X(b\beta)$.
4. If $\gamma = \delta$ accept, else reject.

It is clear that $D$ accepts $(CBC^f, f)$ with probability 1. This is because for any $f$ the following holds. $\alpha = f(a)$; $\beta = f(b)$. $\gamma = CBC^f(a\alpha) = f(f(a) \oplus \alpha) = f(0^n)$. $\delta = CBC^f(b\beta) = f(f(b) \oplus \alpha) = f(0^n) = \gamma$. Also, $D$ accepts $(F, S^F)$ with probability at most $\frac{q^2}{2^n}$, for any simulator $S$ that makes at most $2q$ queries to $F$. This is because of the following reason. The simulator knows $a, b$ and has to come up with $\alpha, \beta$ such that $F(a\alpha) = F(b\beta)$ where $a \neq b$, by making at most $2q$ queries to $F$ (say that it makes at most $q$ queries whose first half is $a$ and at most $q$ queries whose first half is $b$). It is easy to see that for a uniform random function $F$, this happens with probability at most $\frac{q^2}{2^n}$. Hence, we have that for any $S$ that makes at most $2q$ queries to $F$,

$$|Pr\,[D^{CBC^f, f}(1^n) = 1] - Pr\,[D^{F, S^F}(1^n) = 1]| \geq 1 - \frac{q^2}{2^n}$$

Hence we conclude that $D$ distinguishes $(CBC^f, f)$ and $(F, S^F)$ with overwhelming probability for any efficient simulator $S$. □

## 3.2 CBC construction processing inputs of length that are multiples of $n$

Consider the following construction, which is the $CBC$ construction processing inputs of length that are multiples of $n$.

**Definition 3.3.** *Let $f$ be a function such that, for every $n \in \mathbb{Z}^+$, $f : \{0,1\}^n \to \{0,1\}^n$. For every $n \in \mathbb{Z}^+$, define $CBC'^f : (\{0,1\}^n)^* \to \{0,1\}^n$ recursively as follows.*

$$CBC'^f(\epsilon) = 0^n$$

$$CBC'^f(m_1 m_2) = f(CBC'^f(m_1) \oplus m_2)$$

*where $m_1 \in (\{0,1\}^n)^*$, $m_2 \in \{0,1\}^n$ and $\epsilon$ is the empty string.*

It is a well known result that $CBC'^f$ (for a uniform random $f$) does not give us indistinguishability from a uniform random function mapping $(\{0,1\}^n)^*$ to $\{0,1\}^n$ (and hence, it doesn't give us indifferentiability too). However, there are three popular fixes for this construction in the literature, to make the indistinguishability theorem go through. We would like to see if any of these fixes would make the indifferentiability theorem go through for this construction. However, the answer is a negative one as for each of the fixes, we can come up with a distinguisher that is similar to the one in the previous subsection.

### 3.2.1 Variant 1: Prefix-free encoding

The first fix that makes the indistinguishability theorem go through is making sure that the inputs to $CBC'^f$ are not prefixes of each other. We first define what we mean by a prefix-free encoding. We use the definition (and example) from [CDMP05].

**Definition 3.4** (Prefix-free encoding)**.** *Let $n$ be a positive integer. A prefix-free encoding over $\{0,1\}^n$ is an injective function $p : \{0,1\}^* \to (\{0,1\}^n)^*$ such that for every $x, y \in \{0,1\}^*$ where $x \neq y$, $p(x)$ is not a prefix of $p(y)$.*

In practice, we want $p$ to be an easy to compute function. Now, we define a particular prefix-free encoding $p_1$ which will serve as an example, and also be used in a theorem shortly.

**Definition 3.5.** *For every positive integer $n$, define $enc : \{0,1\}^* \to (\{0,1\}^n)^*$ as follows.*

$$enc(m) = m10^x$$

*where $m \in \{0,1\}^*$ and $x \geq 0$ is the least number of zeroes such that $|enc(m)|$ is divisible by $n$.*
*For sufficiently large positive integers $n$, define $p_1 : \{0,1\}^* \to (\{0,1\}^n)^*$ as follows.*

$$p_1(m) = \kappa \; enc(m)$$

*where $m \in \{0,1\}^*$ and $\kappa$ is the $n-$ bit binary encoding of $\frac{|enc(m)|}{n}$* [4].

**Definition 3.6.** *Let $f$ be a function such that, for every $n \in \mathbb{Z}^+$, $f : \{0,1\}^n \to \{0,1\}^n$. Let $p$ be a prefix free encoding over $\{0,1\}^n$, for every $n \in \mathbb{Z}^+$. For every $n \in \mathbb{Z}^+$ define $prefixFreeCBC^{f,p} : \{0,1\}^* \to \{0,1\}^n$ as follows*

$$prefixFreeCBC^{f,p}(m) = CBC'^f(p(m))$$

*where $m \in \{0,1\}^*$.*

[4] Technically, it should be $\frac{|enc(m)|}{n} \bmod 2^n$.

Petrank *et al.* in [PR00] proved that for any prefix-free encoding $p$, the construction $prefixFreeCBC^{f,p}$ (for a uniform random $f$) is indistinguishable from a uniform random function mapping $\{0,1\}^*$ to $\{0,1\}^n$, by extending the proofs in [BKR00]. In [GR10], Gorbunov *et al.* provided an alternate proof for this theorem, by extending the proofs from [Ber05]. In [BPR05], Bellare *et al.* gave another different proof for this theorem, using the so called "Code-based game-playing Framework" and obtained a better quantitative result.

However, it is easy to see that the indifferentiability version of this theorem is not true. That is, there exists a prefix-free encoding $p$ such that $prefixFreeCBC^{f,p}$ (for a uniform random $f$) is not indifferentiable from a uniform random function mapping $\{0,1\}^*$ to $\{0,1\}^n$. The particular prefix-free encoding we consider is $p_1$. The distinguisher (let's call it $D'$) for the proof of this theorem is very similar to the distinguisher $D$ we presented in the previous subsection for the construction $CBC^f$. $D'$ has access to oracles $X : \{0,1\}^* \to \{0,1\}^n$ and $Y : \{0,1\}^n \to \{0,1\}^n$, which are either $(prefixFreeCBC^{f,p_1}, f)$ or $(F, S^F)$. $D'$ works as follows.

1. Choose arbitrary $n$-bit strings $a, b$ such that $a \neq b$.
2. Let $t \leftarrow Y(\overline{3})$ (where $\overline{3}$ is the $n$-bit binary encoding of 3).
3. Let $\alpha \leftarrow Y(t \oplus a)$ and $\beta \leftarrow Y(t \oplus b)$.
4. Let $\gamma \leftarrow X(a\alpha)$ and $\delta \leftarrow X(b\beta)$.
5. If $\gamma = \delta$ accept, else reject.

It is easy to see that $D'$ accepts $prefixFreeCBC^{f,p_1}, f$ with probability 1. This is because
$\gamma = prefixFreeCBC^{f,p_1}(a\alpha) = CBC'^f(p_1(a\alpha)) = CBC'^f(\overline{3}\ a\ \alpha\ 10^{n-1}) = f(f(0^n) \oplus 10^{n-1})$
(since $\alpha = f(f(\overline{3}) \oplus a)$) and
$\delta = prefixFreeCBC^{f,p_1}(b\beta) = CBC'^f(p_1(b\beta)) = CBC'^f(\overline{3}\ b\ \beta\ 10^{n-1}) = f(f(0^n) \oplus 10^{n-1})$
(since $\beta = f(f(\overline{3}) \oplus b)$).
Again, when $D'$ is querying $F, S^F$, the simulator knows $a, b$ and has to come up with $\alpha, \beta$ such that $F(a\alpha) = F(b\beta)$ where $a \neq b$ and a simulator making polynomial number of queries to a uniform random $F$ cannot do this except with negligible probability. We skip the details here, and present just the theorem.

**Theorem 3.7.** *Let $X_n$ be the uniform distribution on the set of all functions mapping $\{0,1\}^*$ to $\{0,1\}^n$ and let $Y_n$ be the uniform distribution on the set $\mathbb{F}_n^n$, for every $n \in \mathbb{Z}^+$. Let $\mathcal{X} = \{X_n\}$ and $\mathcal{Y} = \{Y_n\}$. Then, the construction $prefixFreeCBC^{p_1}$ having oracle access to the distribution family $\mathcal{Y}$ is not indifferentiable from the distribution family $\mathcal{X}$.*

We expect that this theorem is not true for any prefix-free encoding. This is because, we can achieve indifferentiability for $CBC'^f$ (for a uniform random $f$) when we apply it to a very simple encoding that is not even prefix-free (but is injective). The encoding is as follows. Recall the construction $I'^f$ that we introduced in the first section (that is the generalization of the $I^f$). We give an equivalent definition of $I'^f$ here: as $CBC'^f$ applied to a particular encoding.

**Definition 3.8.** *For every positive even integer $n$, define the encoding $e : (\{0,1\}^{n/2})^* \to (\{0,1\}^n)^*$ recursively as follows,*

$$e(\epsilon) = \epsilon$$

$$e(xy) = e(x)y0^{n/2}$$

*where $x \in (\{0,1\}^{n/2})^*$, $y \in \{0,1\}^{n/2}$ and $\epsilon$ is the empty string.*
*Let $f$ be a function such that, for every positive even integer $n$, $f : \{0,1\}^n \to \{0,1\}^n$. For every positive even integer $n$, define $I'^f : (\{0,1\}^{n/2})^* \to \{0,1\}^n$ as follows.*

$$I'^f(m) = CBC'^f(e(m))$$

*where $m \in (\{0,1\}^{n/2})^*$*

We expect that the indifferentiability result holds for $I'^f$ (for a uniform random $f$), though we will only be proving the indifferentiability result for $I^f$ (for a uniform random $f$). As stated in the first section, proving the indifferentiability result for $I'^f$ (for a uniform random $f$) is one of our future goals.

### 3.2.2 Variant 2: Applying a uniform random function, chosen independently of $f$, to the output of $CBC'^f$

The second fix that makes the indistinguishability theorem go through, is to apply a uniform random function (let's call it $g$), chosen independently of $f$, to the output of $CBC'^f$. This construction was introduced in [PR00] as a variant of the CBC construction, that processes inputs of arbitrary length (no prefix-free restriction).

**Definition 3.9.** *Let $f, g$ be functions such that, for every $n \in \mathbb{Z}^+$, $f, g : \{0,1\}^n \to \{0,1\}^n$. For every $n \in \mathbb{Z}^+$, define $doubleCBC^{f,g} : (\{0,1\}^n)^* \to \{0,1\}^n$ as follows.*

$$doubleCBC^{f,g}(m) = g(CBC'^f(m))$$

*where $m \in (\{0,1\}^n)^*$.*

Petrank *et al.* in [PR00] proved that, the construction *doubleCBC*$^{f,g}$ (for $f, g$ chosen independently and uniformly at random) is indistinguishable from a uniform random function mapping $(\{0,1\}^n)^*$ to $\{0,1\}^n$, by extending the proofs in [BKR00].

However, it is easy to see that the indifferentiability version of this theorem is not true. That is, *doubleCBC*$^{f,g}$ (for $f, g$ chosen independently and uniformly at random) is not indifferentiable from a uniform random function mapping $(\{0,1\}^n)^*$ to $\{0,1\}^n$. Again the distinguisher for the proof of this non-differentiability theorem is very similar to the distinguisher $D$ we presented in the previous subsection for the construction $CBC^f$ and the analysis is quite similar too (the simulator's job, while trying to simulate $g$, is to come up with $\alpha, \beta$ such that $F(a\alpha) = F(b\beta)$ where $a \neq b$ and a simulator making polynomial number of queries to a uniform random $F$ cannot do this except with negligible probability). We skip the details, and present just the theorem.

**Theorem 3.10.** *Let $X_n$ be a uniform distribution on the set of all functions mapping $(\{0,1\}^n)^*$ to $\{0,1\}^n$ and let $Y_n$ be a uniform distribution on the set of all 2-tuples of functions where each function maps $\{0,1\}^n$ to $\{0,1\}^n$, for every $n \in \mathbb{Z}^+$. Let $\mathcal{X} = \{X_n\}$ and $\mathcal{Y} = \{Y_n\}$. Then, the construction doubleCBC having oracle access to the distribution family $\mathcal{Y}$ is not indifferentiable from the distribution family $\mathcal{X}$.*

### 3.2.3 Variant 3: The Truncated CBC construction

The final fix we see here is truncating a certain amount of bits (say $\lfloor n/2 \rfloor$ bits) from the output of $CBC'^f$.

**Definition 3.11.** *Let $f$ be a function such that, for every $n \in \mathbb{Z}^+$ such that $n > 1$, $f : \{0,1\}^n \to \{0,1\}^n$. For every $n \in \mathbb{Z}^+$ such that $n > 1$, define $trCBC^f : (\{0,1\}^n)^* \to \{0,1\}^{\lceil n/2 \rceil}$ as follows*

$$trCBC^f(m) = \textit{first } \lceil n/2 \rceil \textit{ bits of } CBC'^f(m)$$

*where $m \in (\{0,1\}^n)^*$.*

It is a well known result that the construction $trCBC^f$ (for uniform random $f$) is indistinguishable from a uniform random function mapping $(\{0,1\}^n)^*$ to $\{0,1\}^{\lceil n/2 \rceil}$. However, the construction $trCBC^f$ (for uniform random $f$) is not indifferentiable from a uniform random function mapping $(\{0,1\}^n)^*$ to $\{0,1\}^{\lceil n/2 \rceil}$. Since $trCBC^f$ can be viewed as being similar to *doubleCBC*$^{f,g}$ where the outer function $g$ is a fixed function that truncates a certain amount of bits, instead of a uniform random function, the same argument as the previous subsection holds here too. Again, we skip the details, and present just the theorem.

**Theorem 3.12.** *Let $X_n$ be the uniform distribution on the set of all functions mapping $(\{0,1\}^n)^*$ to $\{0,1\}^{\lceil n/2 \rceil}$ and let $Y_n$ be the uniform distribution on the set $\mathbb{F}_n^n$, for every $n \in \mathbb{Z}^+$ such that $n > 1$. Let $\mathcal{X} = \{X_n\}$ and $\mathcal{Y} = \{Y_n\}$. Then, the construction trCBC having oracle access to the distribution family $\mathcal{Y}$ is not indifferentiable from the distribution family $\mathcal{X}$.*

**Remark 3.13.** *Note that it doesn't matter how many bits we drop from the output of $CBC^f$ ($\lfloor n/2 \rfloor$ was an arbitrary number). The attack would still go through if say we drop $k$ bits, for any $0 < k < n$, from the output of $CBC^f$.*

**Remark 3.14.** *Note that the distinguisher we described in the proof of Theorem 3.2 is a sequential distinguisher. So, the CBC construction and its variants we described above, do not even satisfy the weaker form of indifferentiability, namely, sequential indifferentiability.*

## 4 Indifferentiability Results for the MD Construction and Some of its Standard Variants

In this section, we present the indifferentiability results for the MD (Merkle-Damgard) construction and some of its standard variants. All the results stated in this section are from [CDMP05]. We restate the results using our notations and definitions. We won't be proving any of the results and suggest the reader to peruse the original paper for the proofs.

### 4.1 MD construction processing 2 blocks of input

We first give the definition of the MD construction processing 2 blocks of input.

**Definition 4.1.** *Let $f$ be a function such that, for every $n \in \mathbb{Z}^+$, $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$. For every $n \in \mathbb{Z}^+$, define $MD^f : \{0,1\}^{2n} \to \{0,1\}^n$ as follows*

$$MD^f(xy) = f(f(0^n, (x)), (y))$$

*where $x, y \in \{0,1\}^n$.*

**Theorem 4.2.** *Let $X_n$ be the uniform distribution on the set $\mathbb{F}_n^{2n}$ and let $Y_n$ be the uniform distribution on the set of all functions mapping $\{0,1\}^n \times \{0,1\}^n$ to $\{0,1\}^n$, for every $n \in \mathbb{Z}^+$. Let $\mathcal{X} = \{X_n\}$ and $\mathcal{Y} = \{Y_n\}$. Then, the construction $MD$ having oracle access to the distribution family $\mathcal{Y}$ is indifferentiable from the distribution family $\mathcal{X}$.*

### 4.2 MD construction processing inputs of length that are multiples of $n$

We now define the more general MD construction which processes inputs of length that are multiples of $n$.

**Definition 4.3.** *Let $f$ be a function such that, for every $n \in \mathbb{Z}^+$, $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$. For every $n \in \mathbb{Z}^+$, define $MD'^f : (\{0,1\}^n)^* \to \{0,1\}^n$ recursively as follows.*

$$MD'^f(\epsilon) = 0^n$$

$$MD'^f(m_1 m_2) = f(MD'^f(m_1), m_2)$$

*where $m_1 \in (\{0,1\}^n)^*$, $m_2 \in \{0,1\}^n$ and $\epsilon$ is the empty string.*

It is a well known result that $MD'^f$, for a uniform random $f$, is indistinguishable from a uniform random function mapping $(\{0,1\}^n)^*$ to $\{0,1\}^n$.

**Theorem 4.4.** *Let $X_n$ be the uniform distribution on the set of all functions mapping $(\{0,1\}^n)^*$ to $\{0,1\}^n$ and let $Y_n$ be the uniform distribution on the set of all functions mapping $\{0,1\}^n \times \{0,1\}^n$ to $\{0,1\}^n$, for every $n \in \mathbb{Z}^+$. Let $\mathcal{X} = \{X_n\}$ and $\mathcal{Y} = \{Y_n\}$. Then, the construction $MD'$ having oracle access to the distribution family $\mathcal{Y}$ is indistinguishable from the distribution family $\mathcal{X}$.*

However, Coron *et al.* in [CDMP05], via a simple attack showed that $MD'^f$, for a uniform random $f$, is not indifferentiable from a uniform random function mapping $(\{0,1\}^n)^*$ to $\{0,1\}^n$. Formally,

**Theorem 4.5.** *Let $X_n$ be the uniform distribution on the set of all functions mapping $(\{0,1\}^n)^*$ to $\{0,1\}^n$ and let $Y_n$ be the uniform distribution on the set of all functions mapping $\{0,1\}^n \times \{0,1\}^n$ to $\{0,1\}^n$, for every $n \in \mathbb{Z}^+$. Let $\mathcal{X} = \{X_n\}$ and $\mathcal{Y} = \{Y_n\}$. Then, the construction $MD'$ having oracle access to the distribution family $\mathcal{Y}$ is not indifferentiable from the distribution family $\mathcal{X}$.*

Coron *et al.* suggested a few ways of modifying the above construction so that the indifferentiability theorem goes through for the modified construction. We present a few of their fixes here.

### 4.2.1 Fix 1: Prefix-free encoding

One way of making the construction work is to ensure that the messages to the $MD'^f$ construction are prefix-free. So, one could employ a suitable prefix free encoding (such as the one presented in the previous section) to the messages before they are processed by the $MD'^f$ construction. Coron *et al.* were able to prove the indifferentiability theorem for such a prefix-free MD construction.

**Definition 4.6.** *Let $f$ be a function such that, for every $n \in \mathbb{Z}^+$, $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$. Let $p$ be a prefix-free encoding over $\{0,1\}^n$, for every $n \in \mathbb{Z}^+$. For every $n \in \mathbb{Z}^+$, define $prefixFreeMD^{f,p} : \{0,1\}^* \to \{0,1\}^n$ as follows*

$$prefixFreeMD^{f,p}(m) = MD'^f(p(m))$$

*where $m \in \{0,1\}^*$.*

**Theorem 4.7.** *Let $X_n$ be the uniform distribution on the set of all functions mapping $\{0,1\}^*$ to $\{0,1\}^n$ and let $Y_n$ be the uniform distribution on the set of all functions mapping $\{0,1\}^n \times \{0,1\}^n$ to $\{0,1\}^n$, for every $n \in \mathbb{Z}^+$. Let $\mathcal{X} = \{X_n\}$ and $\mathcal{Y} = \{Y_n\}$. Then, for every $p$, the construction $prefixFreeMD^p$ having oracle access to the distribution family $\mathcal{Y}$ is indifferentiable from the distribution family $\mathcal{X}$.*

### 4.2.2 Fix 2: Applying a uniform random function, chosen independently from $f$, to the output of the $MD'^f$

Another way of fixing the $MD'^f$ is to apply a random function chosen independently from $f$ to the output of the $MD'^f$ construction.

**Definition 4.8.** *Let $f, g$ be functions such that, for every $n \in \mathbb{Z}^+$, let $f, g : \{0,1\}^n \to \{0,1\}^n$. For every $n \in \mathbb{Z}^+$, define $doubleMD^{f,g} : (\{0,1\}^n)^* \to \{0,1\}^n$ as follows.*

$$doubleMD^{f,g}(m) = g(MD'^f(m))$$

*where $m \in (\{0,1\}^n)^*$.*

**Theorem 4.9.** *Let $X_n$ be the uniform distribution on the set of all functions mapping $(\{0,1\}^n)^*$ to $\{0,1\}^n$ and let $Y_n$ be the uniform distribution on the set of all 2-tuples of functions where each function maps $\{0,1\}^n$ to $\{0,1\}^n$, for every $n \in \mathbb{Z}^+$. Let $\mathcal{X} = \{X_n\}$ and $\mathcal{Y} = \{Y_n\}$. Then, the construction $doubleMD$ having oracle access to the distribution family $\mathcal{Y}$ is indifferentiable from the distribution family $\mathcal{X}$.*

### 4.2.3 Fix 3: The Truncated MD construction

The final fix that we will be presenting here is to drop a fraction (say $\lfloor n/2 \rfloor$) bits from from output of the construction $MD'^f$.

**Definition 4.10.** *Let $f$ be a function such that, for every $n \in \mathbb{Z}^+$ such that $n > 1$, $f : \{0,1\}^n \to \{0,1\}^n$. For every $n \in \mathbb{Z}^+$ such that $n > 1$, define $trMD^f : (\{0,1\}^n)^* \to \{0,1\}^{\lceil n/2 \rceil}$ as follows*

$$trMD^f(m) = \text{first } \lceil n/2 \rceil \text{ bits of } MD'^f(m)$$

*where $m \in (\{0,1\}^n)^*$.*

**Theorem 4.11.** *Let $X_n$ be the uniform distribution on the set of all functions mapping $(\{0,1\}^n)^*$ to $\{0,1\}^{\lceil n/2 \rceil}$ and let $Y_n$ be the uniform distribution on the set $\mathbb{F}_n^n$, for every $n \in \mathbb{Z}^+$ such that $n > 1$. Let $\mathcal{X} = \{X_n\}$ and $\mathcal{Y} = \{Y_n\}$. Then, the construction $trMD$ having oracle access to the distribution family $\mathcal{Y}$ is indifferentiable from the distribution family $\mathcal{X}$.*

Coron *et al.* observed that this construction is not very useful when the output of $MD'^f$ is small (say 128 bits, which is case for the MD5 hash function). This is because, when you drop half the number of bits from the output (say drop 64 bits for a function with 128 bit output), you are left with a function that outputs too few bits to be secure and useful, when used in various protocols. In subsequent work, Maurer *et al.* [MT07] combined the fixes 1 and 3 to get a "truncated prefix-free MD construction" and got a better quantitative indifferentiability result out of it.

## 5 Indifferentiability Results for the Feistel Construction

In this section, we survey some indifferentiability results for the Feistel Construction, which we define below.

**Definition 5.1.** *Let $f$ be a function such that, $f : \{0,1\}^n \to \{0,1\}^n$ for every $n \in \mathbb{Z}^+$. For every $n \in \mathbb{Z}^+$, define $G_f, G_f^{-1} : \{0,1\}^{2n} \to \{0,1\}^{2n}$ (where $G_f^{-1}$ is the inverse of $G_f$) as follows*

$$G_f(LR) = R(L \oplus f(R))$$

$$G_f^{-1}(ST) = (T \oplus f(S))S$$

*where $L, R, S, T \in \{0,1\}^n$*

We now define what we mean by an $i-$ round Feistel network.

**Definition 5.2.** *For every $i \in \mathbb{Z}^+$, let $F^i = \{f_1, ...., f_i\}$ be an $i-$ tuple of functions, where for every $j$, $1 \leq j \leq i$, $f_j : \{0,1\}^n \to \{0,1\}^n$, for every $n$. For every $i \in \mathbb{Z}^+$, for every $n \in \mathbb{Z}^+$, define $\Phi_{i,F_i}, \Phi_{i,F_i}^{-1} : \{0,1\}^{2n} \to \{0,1\}^{2n}$ (where $\Phi_{i,F_i}^{-1}$ is the inverse of $\Phi_{i,F_i}$) as follows*

$$\Phi_{i,F_i}(LR) = G_{f_i}(....(G_{f_2}(G_{f_1}(LR)))....)$$

$$\Phi_{i,F_i}^{-1}(ST) = G_{f_1}^{-1}(....(G_{f_{i-1}}^{-1}(G_{f_i}^{-1}(ST)))....)$$

*where $L, R, S, T \in \{0,1\}^n$*

Luby and Rackoff, in their paper [LR88], showed that the 4-round Feistel network $\Phi_{4,F_4}, \Phi_{4,F_4}^{-1}$, for a uniform random $F_4$, where each function maps $\{0,1\}^n$ to $\{0,1\}^n$ (by uniform random $F_4$, we refer to a 4-tuple of functions chosen uniformly at random from all possible 4-tuples of functions), is indistinguishable from a uniform random permutation, together with its inverse, mapping $\{0,1\}^{2n}$ to $\{0,1\}^{2n}$.

However, Coron *et al.*, in [CPS08], showed that even the 5-round Feistel network $\Phi_{5,F_5}, \Phi_{5,F_5}^{-1}$, for a uniform random $F_5$ (where each function maps $\{0,1\}^n$ to $\{0,1\}^n$), is not indifferentiable from a uniform random permutation, together with its inverse, mapping $\{0,1\}^{2n}$ to $\{0,1\}^{2n}$. We restate the theorem here, using our notations and definitions. Recall that for every $n \in \mathbb{Z}^+$, $P_n$ denotes the set of all permutations (together with their inverses) mapping $\{0,1\}^n$ to $\{0,1\}^n$.

**Theorem 5.3.** *Let $X_n$ be the uniform distribution on the set $\mathbb{P}_{2n}$ and let $Y_n$ be the uniform distribution on the set of all 5-tuples of functions where each function maps $\{0,1\}^n$ to $\{0,1\}^n$, for every $n \in \mathbb{Z}^+$. Let $\mathcal{X} = \{X_n\}$ and $\mathcal{Y} = \{Y_n\}$. Then, the 5-round Feistel network $\Phi_5, \Phi_5^{-1}$ having oracle access to the distribution family $\mathcal{Y}$ is not indifferentiable from the distribution family $\mathcal{X}$.*

To prove the theorem, they gave a distinguisher $D$ and claimed that $D$ that distinguished between $((\Phi_{5,F_5}, \Phi_{5,F_5}^{-1}), F_5)$ and $((P, P^{-1}), S^{P,P^{-1}})$ with overwhelming probability for any efficient simulator $S$, where $F_5$ is a uniform random 5-tuple of functions (where each function maps $\{0,1\}^n$ to $\{0,1\}^n$), $(P, P^{-1})$ is a uniform random permutation and its inverse, mapping $\{0,1\}^{2n}$ to $\{0,1\}^{2n}$. However, we have found a minor technical flaw in this proof which makes their claim incorrect. We refer the reader to the appendix for a detailed explanation the flaw and the suggested fix.

Mandal *et al.*, in [MPS12], showed that the 6-round Feistel network $\Phi_{6,F_6}, \Phi_{6,F_6}^{-1}$, for a uniform random $F_6$ (where each function maps $\{0,1\}^n$ to $\{0,1\}^n$), is sequentially indifferentiable from a uniform random permutation, together with its inverse, mapping $\{0,1\}^{2n}$ to $\{0,1\}^{2n}$. We restate the theorem here, using our notations and definitions.

**Theorem 5.4.** *Let $X_n$ be the uniform distribution on the set $\mathbb{P}_{2n}$ and let $Y_n$ be the uniform distribution on the set of all 6-tuples of functions where each function maps $\{0,1\}^n$ to $\{0,1\}^n$, for every $n \in \mathbb{Z}^+$. Let $\mathcal{X} = \{X_n\}$ and $\mathcal{Y} = \{Y_n\}$. Then, the 6-round Feistel network $\Phi_6, \Phi_6^{-1}$ having oracle access to the distribution family $\mathcal{Y}$ is sequentially indifferentiable from the distribution family $\mathcal{X}$.*

Coron *et al.*, in [CPS08], showed that the 6-round Feistel network $\Phi_{6,F_6}, \Phi_{6,F_6}^{-1}$, for a uniform random $F_6$ (where each function maps $\{0,1\}^n$ to $\{0,1\}^n$), is indifferentiable from a uniform random permutation, together with its inverse, mapping $\{0,1\}^{2n}$ to $\{0,1\}^{2n}$. However, this was shown to be false in [HKT10] by Holenstein et al, where they gave a distinguisher $D$ that distinguished between $((\Phi_{6,F_6}, \Phi_{6,F_6}^{-1}), F_6)$ and $((P, P^{-1}), S^{P,P^{-1}})$ with overwhelming probability, where $F_6$ is a uniform random 6-tuple of functions (where each function maps $\{0,1\}^n$ to $\{0,1\}^n$), $(P, P^{-1})$ is a uniform random permuation and its inverse, mapping $\{0,1\}^{2n}$ to $\{0,1\}^{2n}$ and $S^{P,P^{-1}}$ is the simulator defined in [CPS08].

Holenstein *et al.*, in [HKT10], showed that the 14-round Feistel network $\Phi_{14,F_{14}}, \Phi_{14,F_{14}}^{-1}$, for a uniform random $F_{14}$ (where each function maps $\{0,1\}^n$ to $\{0,1\}^n$), is indifferentiable from a uniform random permutation, together with its inverse, mapping $\{0,1\}^{2n}$ to $\{0,1\}^{2n}$. We restate the theorem here, using our notations and definitions.

**Theorem 5.5.** *Let $X_n$ be the uniform distribution on the set $\mathbb{P}_{2n}$ and let $Y_n$ be the uniform distribution on the set of all 14-tuples of functions where each function maps $\{0,1\}^n$ to $\{0,1\}^n$, for every $n \in \mathbb{Z}^+$. Let $\mathcal{X} = \{X_n\}$ and $\mathcal{Y} = \{Y_n\}$. Then, the 14-round Feistel network $\Phi_{14}, \Phi_{14}^{-1}$ having oracle access to the distribution family $\mathcal{Y}$ is indifferentiable from the distribution family $\mathcal{X}$.*

**Remark 5.6.** *Note that the distinguisher of [HKT10] does not prove that the 6-round Feistel network, for a uniform random $F_6$ (where each function maps $\{0,1\}^n$ to $\{0,1\}^n$), is not indifferentiable from a uniform random permutation, together with its inverse, mapping $\{0,1\}^{2n}$ to $\{0,1\}^{2n}$. It just shows that the simulator of [CPS08] is incorrect. Currently it is not known if the $i-$round Feistel Network (for a uniform random $F_i$) for each $i$, $6 \leq i \leq 13$ is indifferentiable from a uniform random permutation, together with its inverse, mapping $\{0,1\}^{2n}$ to $\{0,1\}^{2n}$.*

**Remark 5.7.** *Note that in the indifferentiability setting for an $i-$round Feistel network (say $i \geq 6$), to have any hope of achieving indifferentiability, it is quite necessary that the simulator be given access to the inverse permutation oracle. This is because of the inherent invertibility of Feistel in the indifferentiability setting: since the adversary has access to the $i$ round functions, it can do inverse permutation operations by just querying the $i$ round functions (so the adversary doesn't need the inverse oracle).*

## 6 Indistinguishability Proof of $I^f$, for a Uniform Random $f$

Consider the following construction (which we introduced in the first section and restate it here).

**Definition 6.1.** *Let $f$ be a function such that, for every positive even integer $n$, $f : \{0,1\}^n \to \{0,1\}^n$. For every positive even integer $n$, define $I^f : \{0,1\}^n \to \{0,1\}^n$ as follows*

$$I^f(xy) = f(f(x0^{n/2}) \oplus y0^{n/2})$$

*where $x, y \in \{0,1\}^{n/2}$.*

This construction closely resembles the sponge construction [BDPVA11a], on which the winner of SHA-3 competition [BDPVA11b] is based. Our main goal is to give an easy to understand indifferentiability proof for this construction using a simple and rigorous technique. But first, we give an indistinguishability proof for this construction. Since this construction also resembles CBC, one would expect that the easiest way to prove the indistinguishability result for this construction, is using the technique that Bernstein uses to prove the security of CBC in [Ber05]. However, it turns out that we can get a much simpler and more intuitive proof, using a technique called coupling [Lin92][Tho00] from probability theory.

We now proceed to use coupling to prove that the construction $I^f$, for a uniform random a $f$, is indistinguishable from a uniform random function $F : \{0,1\}^n \to \{0,1\}^n$.

We only consider distinguishers that make exactly $q$ queries to an oracle that is either $I^f$ or $F$. It is not difficult to see that this can be assumed without loss of generality. We fix $n$ to be a sufficiently large positive even integer and $q$ to be a polynomial in $n$. Now we proceed to prove the following theorem which would directly imply the main indistinguishability theorem (which we will be stating after the proof of this theorem).

**Theorem 6.2.** *Let $f, F$ be functions chosen independently and uniformly at random from all functions mapping $\{0,1\}^n$ to $\{0,1\}^n$ . Let the construction $I^f$ be as defined before. Then for any distinguisher $D$ that makes $q$ queries to an oracle that is either $I^f$ or $F$, we have that*

$$|Pr\,[D^{I^f}(1^n) = 1] - Pr[D^F(1^n) = 1]| \le \frac{1}{2}\frac{q(q+1)}{2^{n/2}}$$

*Proof.* We fix $D$. Without loss of generality, we can assume that the distinguisher $D$ is deterministic and never repeats a query.

We define Games $G_1, G_2$ (Figure 1) such that they correspond to $D$ interacting with the distributions $F$ and $I^f$ respectively. For the sake of clarity, we introduce an oracle $A$ in each game such that $D$ only queries $A$ in each game, and $A$ queries either $F$ or computes $I^f$ by making two queries to $f$, depending on the game. We choose $f, F$ independently and uniformly at random from all functions mapping $\{0,1\}^n$ to $\{0,1\}^n$ and run Games $G_1$ and $G_2$ using $f, F$ as follows.

- **Game** $G_1$. In Game $G_1$, on $i^{th}$ query $uv$ (where $1 \le i \le q$ and $u, v \in \{0,1\}^{n/2}$) to $A$ by $D$, $A$ returns $F(uv)$.
- **Game** $G_2$. In Game $G_2$, on the $i^{th}$ query $uv$ (where $1 \le i \le q$ and $u, v \in \{0,1\}^{n/2}$) to $A$ by $D$, $A$ computes $I^f(uv)$ by making two queries to $f$ and returns the result.

In both $G_1$ and $G_2$, we abuse the notation $A(uv)$ to refer to two things. When we say query $A(uv)$, we refer to a query $uv$ made by $D$ to $A$, and when $A(uv)$ is preceded or followed by an assignment or $\oplus$ operation, we refer to the result of a query $uv$ made by $D$ to $A$. Also, the notation $A_j(uv)$ (where $1 \le j \le q$) refers to the $j^{th}$ query $uv$ made by $D$ or the output of the $j^{th}$ query $uv$ made by $D$, depending on the sense in which it is used. Here $u, v \in \{0,1\}^{n/2}$. Also in Game $G_2$, we abuse the notation $f(uv)$ to refer to two things.

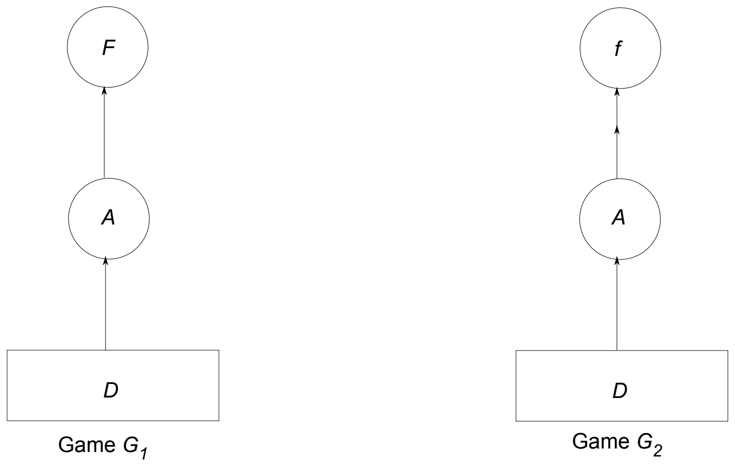

Figure 1: Illustrations representing Games $G_1$ and $G_2$. In Game $G_2$, the double arrow from $A$ to $f$ denotes that, on each query made to $A$ by $D$, $A$ makes two queries to $f$.

When we say query $f(uv)$, we refer to a query $uv$ made to $f$, and when $f(uv)$ is preceded or followed by an assignment or $\oplus$ operation, we refer to the the result of a query $uv$ made to $f$. Here $u, v \in \{0,1\}^{n/2}$.

For each $i \in \{1,2\}$, let $p_i$ be the probability that $D$ accepts in Game $G_i$ (The probability $p_1$ is over the choice of $F$ and the probability $p_2$ is over the choice of $f$). We would like to show to show that $|p_2 - p_1|$ is small, that is, at most $\frac{1}{2}\frac{q(q+1)}{2^{n/2}}$.

The intuition is as follows. In the Game $G_1$ where $D$ queries $A$, and $A$ queries $F$, the queries made to $F$ are answered randomly. In the Game $G_2$ where $D$ queries $A$, and $A$ computes $I^f$, on a query $uv$ (where $u, v \in \{0,1\}^{n/2}$) to $A$, $A$ makes two queries to $f$ and returns the result of the second query $f(f(u0^{n/2}) \oplus v0^{n/2})$. Now, as long as the query $f(f(u0^{n/2}) \oplus v0^{n/2})$ is a query that has not been made to $f$ before, the result of the query will be a value chosen uniformly at random. We will argue that with high probability, this is indeed the case and hence say that with high probability, the outcomes (that is, the answer to the queries that $D$ makes) of both the games are distributed identically.

Here is a rigorous way to argue this. We *couple* the games by running both the games together using common randomness (Figure 2) and argue that with high probability, the outcomes of the coupled games are identical. This is a well known technique from probability theory, called coupling [Lin92][Tho00]. Consider the following experiment.

**BEGIN EXPERIMENT**

- **Initialization-** Choose $2q$ strings $r_1, r_2, ..., r_q, s_1, s_2, ..., s_q$ randomly from $\{0,1\}^n$. For every $i$, $1 \le i \le q$, let $s_i = s_i^1 s_i^2$ where $|s_i^1| = |s_i^2| = n/2$.
- We run the Game $G_1$ using the the $2q$ strings as follows. On the $i^{th}$ query $uv$ (where $1 \le i \le q$ and $u, v \in \{0,1\}^{n/2}$) by $A$, we return $r_i$. Let us call this (slightly) modified game as $G_1'$. Let $p_1'$ be the probability that the $D$ accepts in Game $G_1'$. It is easy to see that the $p_1' = p_1$.

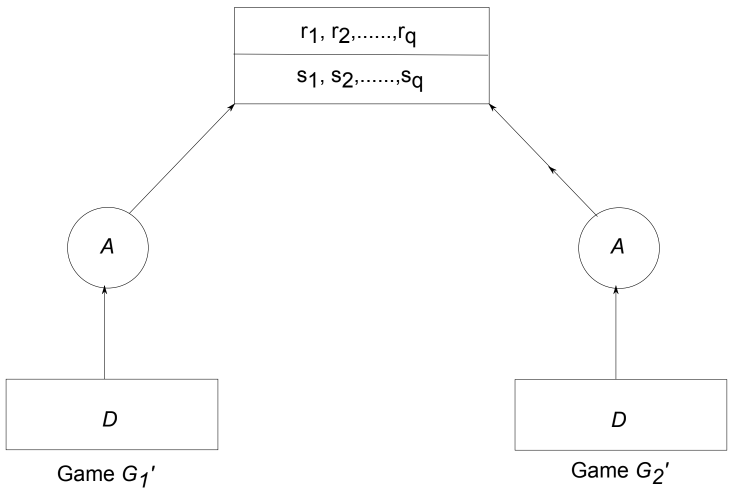

Figure 2: Illustrations representing Games $G_1'$ and $G_2'$. In Game $G_2'$, the double arrow from $A$ to $f$ denotes that, on each query made to $A$ by $D$, $A$ makes two queries to $f$.

- We run the Game $G_2$ using the the $2q$ strings as follows. Intuitively, on a query $A_i(uv)$ (where $1 \leq i \leq q$ and $u, v \in \{0,1\}^{n/2}$) we would like to simulate $f(f(u0^{n/2}) \oplus v0^{n/2})$ using the $2q$ strings by returning $s_i$ for the first call made by $A$, and $r_i$ for the second call made by $A$, as long as the calls are new. Here is how we do this. On a query $A_i(uv)$, if the first call (that is, $u0^{n/2}$) made by $A$ is a new call, we return $s_i$. Else, we return what we returned before when the call was previously made. Let $S_i$ be the value returned. If the second call (that is, $S_i \oplus v0^{n/2}$) made by $A$ is a new call, we return $r_i$. Else, we return what we returned before when the call was previously made. Let us call this (slightly) modified game as $G_2'$. Let $p_2'$ be the probability that the $D$ accepts in Game $G_2'$. It is not difficult to see that the $p_2' = p_2$.

**END EXPERIMENT**

Since $p_1' = p_1$ and $p_2' = p_2$, it is sufficient to show that $|p_2' - p_1'|$ is small, that is, at most $\frac{1}{2}\frac{q(q+1)}{2^{n/2}}$. We do this as follows.

We define the event $B$ that can happen in Game $G_2'$. Intuitively, in Game $G_2'$ an event $B$ represents the scenarios in which $r_i$ is not returned on the query $A_i(uv)$ made by $D$ for some $1 \leq i \leq q$, $u, v \in \{0,1\}^{n/2}$. That is, it represents the scenarios in which the second call made by $A$ is a previous first call made by $A$, or a previous second call made by $A$ (and hence is not a new call).

**Definition 6.3** (Event $B$)**.** *In Game $G_2'$, we say that an event $B$ has happened if the following occurs.*

- *We have queries $A_i(uv)$, $A_t(u'v')$ for some $1 \leq t, i \leq q$, $u, v, u', v' \in \{0,1\}^{n/2}$ and*
  *$t \leq i$ and $S_i \oplus v0^{n/2} = u'0^{n/2}$ or*
  *$t < i$ and $S_i \oplus v0^{n/2} = S_t \oplus v'0^{n/2}$.*

Note that in the above definition, if we have queries $A_i(uv)$, $A_t(u'v')$ for some $1 \leq t < i \leq q$, $u, v, u', v' \in \{0,1\}^{n/2}$ and $S_i \oplus v0^{n/2} = S_t \oplus v'0^{n/2}$, then we have that $u \neq u'$. This is due the following reason. Suppose $u = u'$. Then by the definition of Game $G_2'$, we have that $S_i = S_t$. Since we have that $S_i \oplus v0^{n/2} = S_t \oplus v'0^{n/2}$, we would get that $v = v'$ and hence $uv = u'v'$. But this would contradict the assumption that the queries made by $D$ are distinct.

Now, we define the event $B'$ that can occur in the **Initialization** step. The intuition behind defining $B'$ is as follows.

1. It will turn out that if $B'$ does not occur, then the event $B$ (in Game $G'_2$) does not occur.
2. We will see that the probability that $B'$ occurs (over the choice of $s_1, ..., s_q$) will be easy to calculate and is small.

**Definition 6.4** (Event $B'$)**.** *In the **Initialization** step, we say that an event $B'$ has happened if the following occurs.*

- *For some $i, j$ s.t. $1 \leq i \neq j \leq q$, we have that*
  *$s_i^2 = 0^{n/2}$ or*
  *$s_i^2 = s_j^2$*

First, we prove a lemma regarding what the value of $S_i$ will be (for every $i$, $1 \leq i \leq q$) in Game $G'_2$. Intuitively, for a query $A_i(uv)$ (for some $u, v \in \{0,1\}^{n/2}$), $S_i$ will be equal to $s_j$ for some $1 \leq j \leq i$ such that the the $j^{th}$ $A$ query is the query where $u$ appears for the first time as the first half of an $A$ query.

**Lemma 6.5.** *Assume $B'$ does not occur in the Initialization step. Then in Game $G'_2$, for every $i$, $1 \leq i \leq q$, the following holds.*

*Consider the query $A_i(uv)$ (for some $u, v \in \{0,1\}^{n/2}$). Let the query $A_j(uv')$ (for some $j$, $1 \leq j \leq i$ and for some $v' \in \{0,1\}^{n/2}$) be such that there exists no query of the form $A_k(uv'')$ (for some $k$, $1 \leq k < j$ and for some $v'' \in \{0,1\}^{n/2}$). Then, we have that $S_i = s_j$.*

*Proof.* We prove this lemma using complete induction on $i$.

Say $1 \leq i \leq q$ and for all $l$, $1 \leq l < i$, the following holds. Consider the query $A_l(uv)$ (for some $u, v \in \{0,1\}^{n/2}$). Let the query $A_m(uv')$ (for some $m$, $1 \leq m \leq l$ and for some $v' \in \{0,1\}^{n/2}$) be such that there exists no query of the form $A_n(uv'')$ (for some $n$, $1 \leq n < m$ and for some $v'' \in \{0,1\}^{n/2}$). Then, we have that $S_l = s_m$.

We want to show the following. Consider the query $A_i(uv)$ (for some $u, v \in \{0,1\}^{n/2}$). Let the query $A_j(uv')$ (for some $j$, $1 \leq j \leq i$ and for some $v' \in \{0,1\}^{n/2}$) be such that there exists no query of the form $A_k(uv'')$ (for some $k$, $1 \leq k < j$ and for some $v'' \in \{0,1\}^{n/2}$). Then, we have that $S_i = s_j$.

On the query $A_i(uv)$, the first call made by $A$ is $u0^{n/2}$. This can either be a new call or not. We consider both the cases.

1. We have that $u0^{n/2}$ is a new call made by $A$. Hence, by the definition of Game $G'_2$, we have that $S_i = s_i$. Hence, by induction, the lemma holds for this case.
2. We have that $u0^{n/2}$ is not a new call made by $A$. Now, this call could have been made as a previous first call or a previous second call by $A$. If it was a previous first call by $A$, then we have that $S_i = S_k$ for some $1 \leq k < i$. But by induction hypothesis, we have that $S_k = s_j$ for some $1 \leq j \leq k$ where the query $A_j(uv')$ (for some $v' \in \{0,1\}^{n/2}$) is such that there exists no query of the form $A_k(uv'')$ (for some $k$, $1 \leq k < j$ and for some $v'' \in \{0,1\}^{n/2}$). Hence, we have that $S_i = s_j$. Hence, by induction, the lemma holds for this sub-case. If it was a previous second call by $A$ for some query $A_k(st)$ (for some $k$, $1 \leq k < i+1$ and for some $s, t \in \{0,1\}^{n/2}$), then we have that $u0^{n/2} = S_k \oplus t0^{n/2}$. But, this must mean that the last $n/2$ bits of $S_k$ is $0^{n/2}$. By induction hypothesis, we have that $S_k = s_j$ for some $1 \leq j \leq k$ where the query $A_j(sv')$ (for some $v' \in \{0,1\}^{n/2}$) is such that there exists no query of the form $A_k(sv'')$ (for some $k$, $1 \leq k < j$ and for some $v'' \in \{0,1\}^{n/2}$). Hence, this would mean that $s_j^2 = 0^{n/2}$ and hence $B'$ would have occurred in the **Initialization** step.

□

Now, we proceed to prove the following lemma.

**Lemma 6.6.** *If the event $B'$ does not occur, then the event $B$ does not occur.*

*Proof.* Suppose event $B$ has occurred (in Game $G_2$). So, this means that we have queries $A_i(uv)$, $A_t(u'v')$ for some $1 \leq t, i \leq q$, $u, v, u', v' \in \{0,1\}^{n/2}$ and
$t \leq i$ and $S_i \oplus v0^{n/2} = u'0^{n/2}$ or
$t < i$ and $S_i \oplus v0^{n/2} = S_t \oplus v'0^{n/2}$.

If $t \leq i$ and $S_i \oplus v0^{n/2} = u'0^{n/2}$, then we have that the last $n/2$ bits of $S_i$ is $0^{n/2}$. Now, according to the lemma 6.5, $S_i = s_j$ for some $1 \leq j \leq i$ where the query $A_j(uw')$ (for some $w' \in \{0,1\}^{n/2}$) is such that there exists no query of the form $A_k(uw'')$ (for some $k$, $1 \leq k < j$ and for some $w'' \in \{0,1\}^{n/2}$). So, we have that $s_j^2 = 0^{n/2}$ for some $1 \leq j \leq i$. Hence, we have that event $B'$ had occurred (in the **Initialization** step).

If $t < i$ and $S_i \oplus v0^{n/2} = S_t \oplus v'0^{n/2}$ (and thereby $u \neq u'$), then we have that the last $n/2$ bits of $S_i$ and $S_t$ are equal. By Lemma 6.5, we have that $S_i = s_k$ for some $1 \leq k \leq i$ where the query $A_k(uw')$ (for some $w' \in \{0,1\}^{n/2}$) is such that there exists no query of the form $A_p(uw'')$ (for some $p$, $1 \leq p < k$ and for some $w'' \in \{0,1\}^{n/2}$). Also, by Lemma 6.5, we have that $S_t = s_l$ for some $1 \leq l \leq t$ where the query $A_l(u'w')$ (for some $w' \in \{0,1\}^{n/2}$) is such that there exists no query of the form $A_p(u'w'')$ (for some $p$, $1 \leq p < l$ and for some $w'' \in \{0,1\}^{n/2}$). Also, since $S_i \neq S_t$ (because $u \neq u'$), we have that $k \neq l$. So, we have that $s_k^2 = s_l^2$ where $k \neq l$. Hence, we have that event $B'$ occurs had occurred (in the **Initialization** step). □

Now, we prove that for most choices of $s_1, .., s_q$, the event $B'$ does not occur.

**Lemma 6.7.** $Pr[B'] \leq \frac{1}{2}\frac{q(q+1)}{2^{n/2}}$. *The probability is over the choice of $s_1, ..., s_q$.*

*Proof.* In the set $s_1, ..., s_q$, the probability that $s_i^2 = 0^{n/2}$ or $s_i^2 = s_j^2$ for some $i, j$, $1 \leq i \neq j \leq q$ is at most $\frac{q+\binom{q}{2}}{2^{n/2}}$. Hence, we have that $Pr[B'] \leq \frac{1}{2}\frac{q(q+1)}{2^{n/2}}$. □

Now, we conclude the proof of the theorem as follows.

**Lemma 6.8.** *Given that the event $B$ does not happen (in Game $G_2'$), in both the games $G_1'$ and $G_2'$, for every $i$, $1 \leq i \leq q$, the result of the $i^{th}$ query by $D$ is $r_i$. Therefore, $D$ accepts in Game $G_1'$ iff it accepts in Game $G_2'$.*

*Proof.* By the definition of Game $G_1'$, for every $i$, $1 \leq i \leq q$, the result of the $i^{th}$ query by $D$ is $r_i$. In Game $G_2'$, as long as the event $B$ does not occur, the second query made by $A$ is a new query. Hence, by the definition of Game $G_2'$, for every $i$, $1 \leq i \leq q$, the result of the $i^{th}$ query by $D$ is $r_i$. Since $D$ sees identical replies in both the games, it is easy to see that $D$ accepts in Game $G_1'$ iff it accepts in Game $G_2'$. □

Hence, by Lemma 6.8 we have that $|p_2' - p_1'| \leq Pr[B]$. By Lemma 6.6, we have that $Pr[B] \leq Pr[B']$. By Lemma 6.7, we have that $Pr[B'] \leq \frac{1}{2}\frac{q(q+1)}{2^{n/2}}$. Hence, we have that $|p_2' - p_1'| \leq \frac{1}{2}\frac{q(q+1)}{2^{n/2}}$.

Hence, we conclude that $|p_2 - p_1| \leq \frac{1}{2}\frac{q(q+1)}{2^{n/2}}$. □

We now state the indistinguishability theorem about the construction $I^f$, for a uniform random $f$. The proof of the theorem directly follows from Theorem 6.2.

**Theorem 6.9.** *Let $X_n$ be the uniform distributions on $\mathbb{F}_n^n$, for every positive even integer $n$. Let $\mathcal{X} = \{X_n\}$. Then, construction $I$ having oracle access to the distribution family $\mathcal{X}$ is indistinguishable from the distribution family $\mathcal{X}$.*

*Proof.* Directly follows from Theorem 6.2. □

## 6.1 Generalization to $k$ blocks of input

It is not difficult to see that the above proof can be generalized to get indistinguishability for the following construction.

**Definition 6.10.** *Let $k$ be a positive integer. Let $f$ be a function such that $f : \{0,1\}^n \to \{0,1\}^n$, for every positive even integer $n$. For every positive even integer $n$, define $I_1^f : (\{0,1\}^{n/2})^k \to \{0,1\}^n$ as follows*

$$I_1^f(x_1x_2....x_k) = f(f(.....f(f(x_1 0^{n/2}) \oplus x_2 0^{n/2}) \oplus x_3 0^{n/2}.....) \oplus x_k 0^{n/2})$$

*where $x_1, x_2, ..., x_k \in \{0,1\}^{n/2}$.*

We state the following theorem about $I_1^f$, for a uniform random $f$ (without giving the proof), which would imply the indistinguishability theorem about $I_1^f$, for a uniform random $f$. We fix $k$. We fix $n$ to be a sufficiently large positive even integer and $q$ to be a polynomial in $n$.

**Theorem 6.11.** *Let $f$ be a function chosen uniformly at random from all functions mapping $\{0,1\}^n$ to $\{0,1\}^n$. Let $F$ be a function chosen uniformly at random from all functions mapping from $(\{0,1\}^{n/2})^k$ to $\{0,1\}^n$. Let the construction $I_1^f$ be as defined before. Then, for any distinguisher $D$ that makes $q$ queries to an oracle that is either $I_1^f$ or $F$, we have that*

$$|Pr\ [D^{I_1^f}(1^n) = 1] - Pr[D^F(1^n) = 1]| \le \frac{1}{2}\frac{(k-1)(q^2-q)+2q}{2^{n/2}}$$

# 7 Indifferentiability Proof of $I^f$, for a Uniform Random $f$

We now proceed to prove that the construction $I^f$, for a uniform random $f : \{0,1\}^n \to \{0,1\}^n$, is indifferentiable from a uniform random function $F : \{0,1\}^n \to \{0,1\}^n$, and in the process give a detailed explanation of the technique we use for proving it. We restate the definition of the construction $I$.

**Definition 7.1.** *Let $f$ be a function such that, for every positive even integer $n$, $f : \{0,1\}^n \to \{0,1\}^n$. For every positive even integer $n$, define $I^f : \{0,1\}^n \to \{0,1\}^n$ as follows*

$$I^f(xy) = f(f(x0^{n/2}) \oplus y0^{n/2})$$

*where $x, y \in \{0,1\}^{n/2}$.*

We have to prove that the construction $I^f$, for a uniform random $f : \{0,1\}^n \to \{0,1\}^n$, is indifferentiable from a uniform random function $F : \{0,1\}^n \to \{0,1\}^n$. To this effect, we define an efficient simulator $S$ that has oracle access to $F$ and simulates $f$ (in response to a polynomial number of queries made to it), such that no distinguisher that makes a polynomial number of queries to a pair of oracles, that is either $(I^f, f)$ or $(F, S^F)$, can distinguish between them, except with negligible probability.

**Overview of Simulator.** We let the simulator $S$ to be a deterministic algorithm as follows- in addition to having oracle access to $F$, we let the simulator have oracle access to a random function $f$ that it queries whenever it has to make a random choice. From here onwards, we use the notation $S^{F,f}$ to refer to the simulator, since it has oracle access to functions $F, f$. The simulator $S^{F,f}$ maintains a history of the the queries it has received and its corresponding responses so far, in form of a set as defined below.

$$H^S : \{(\alpha, \beta) \in \{0,1\}^n \times \{0,1\}^n \mid \text{the simulator has responded with}\, \beta \,\text{to a query}\, \alpha\}$$

$H^S$ is initially empty. On a query $uv$ (where $u, v \in \{0,1\}^{n/2}$), the simulator calls an internal procedure Query($uv$). The procedure first checks if $\exists z$ such that, $(uv, z) \in H^S$ (that is, the simulator had previously been queried $uv$ and had returned $z$). If yes, the procedure returns $z$. This is captured by lines $3-4$ of

Algorithm 1. If that is not the case, the procedure checks if $\exists a, b \in \{0,1\}^{n/2}$ such that, $(a0^{n/2}, bv) \in H^S$. If that is the case, the procedure makes an oracle query $(a(b \oplus u))$ to $F$. Let $z$ be the result of the oracle query. The procedure adds $(uv, z)$ to the history $H^S$ and returns $z$. This is captured by lines $7 - 11$ of Algorithm 1. If that is not the case, the procedure makes calls the function $f$ on $uv$. Let $z$ be the value returned. The procedure stores $(uv, z)$ to the history $H^S$ and returns $z$. This is captured by lines $13 - 15$ of Algorithm 1. Finally, the simulator returns whatever the procedure returns. That is, the simulator $S^{F,f}$ on a new input, queries either $F$ (line 9 of Algorithm 1) or $f$ (line 13 of Algorithm 1) and adds the input-output pair (the input to the simulator and the output of the simulator's query to either $F$ or $f$) to $H^S$ before returning the output. This makes $H^S$ a binary relation between $\{0,1\}^n$ and $\{0,1\}^n$. However, $H^S$ is also a partial function from $\{0,1\}^n$ to $\{0,1\}^n$, because of lines $3 - 4$ of Algorithm 1.

The intuition here is that the simulator keeps returning random values (using $f$) unless it detects that there has already been a query $a0^{n/2}$ (for some $a \in \{0,1\}^{n/2}$) such that it had replied with $bv$ for some $b \in \{0,1\}^{n/2}$ (that is, the value it had replied with has the same last $n/2$ bits as the current query). In that case, it returns $F(a(b \oplus u))$ (here $uv$ is the current query where $u, v \in \{0,1\}^{n/2}$). The idea here is to return a value 'consistent' with the construction $I^f$, so that no distinguisher that makes a polynomial number of queries to a pair of oracles, that is either $(I^f, f)$ or $(F, S^{F,f})$, can distinguish between them, except with negligible probability.

**Algorithm 1** Simulator $S^{F,f}$

```
1:  variable: Simulator history $H^S$
2:  procedure QUERY($uv$)
3:      if $\exists z$ s.t. $(uv, z) \in H^S$ then
4:          return $z$
5:      else
6:          if $\exists a, b \in \{0,1\}^{n/2}$ s.t. $(a0^{n/2}, bv) \in H^S$ then
7:              Choose the first such $(a, b)$ pair    ▷ We will see that for every $D$, with high probability (over
    the randomness of $F, f$), there will exist a unique $(a, b)$ pair.
8:              Let $z \leftarrow F(a(b \oplus u))$
9:              Add $(uv, z)$ to $H^S$
10:             return $z$
11:         else
12:             Let $z \leftarrow f(uv)$
13:             Add $(uv, z)$ to $H^S$
14:             return $z$
15:         end if
16:     end if
17: end procedure
```

In the discussions below, we refer a query to $I^f$ or $F$ as a query to the left oracle $L$ and a query to $f$ or $S^{F,f}$ as a query to the right oracle $R$. We only consider the distinguishers that make exactly $q$ queries to the Left oracle $L$ and exactly $q$ queries to the Right oracle $R$ and the queries are of the form an $L$ query, an $R$ query, an $L$ query, an $R$ query and so on (so the distinguisher totally makes $2q$ queries). It is not difficult to see that this can be assumed without loss of generality. We fix $n$ to be a sufficiently large positive even integer and $q$ to be a polynomial in $n$. Now we proceed to prove the following theorem which would directly imply the main indifferentiability theorem (which we will be stating after the proof of this theorem).

**Theorem 7.2.** *Let $f, F$ be functions chosen independently and uniformly at random from all functions mapping $\{0,1\}^n$ to $\{0,1\}^n$. Let the construction $I^f$ be as defined before. Let $S^{F,f}$ be the simulator as defined above. Then for any distinguisher $D$ that makes $q$ queries to $L$ and $q$ queries to $R$ in the manner specified*

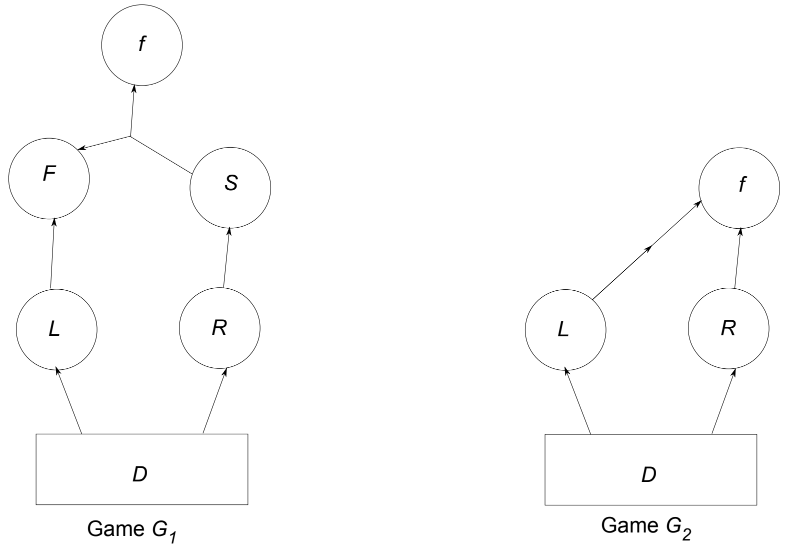

Figure 3: Illustrations representing Games $G_1$ and $G_2$. In Game $G_1$, the arrow from $S$ which branches to $F$ and $f$ denotes that, on each query made to the simulator by $R$, the simulator queries either $F$ or $f$. In Game $G_2$, the double arrow from $L$ to $f$ denotes that, on each query made to $L$ by $D$, $L$ makes two queries to $f$.

*above, we have that*

$$|Pr\ [D^{I^f,f}(1^n) = 1] - Pr[D^{F,S^F}(1^n) = 1]| \leq \frac{1}{2}\frac{7q^2+3q}{2^{n/2}} + \frac{q}{2^n}$$

Note that the simulator $S^{F,f}$ makes at most $q$ oracle queries to $F$ and runs in time $t_S = O(q)$.

*Proof.* We fix $D$. Without loss of generality, we can assume that the distinguisher $D$ is deterministic and never repeats a query to the Left oracle $L$ and never repeats a query to the Right oracle $R$.

Now we define Games $G_1, G_2$ (Figure 3) such that they correspond to $D$ interacting with $(F, S^{F,f})$ and $(I^f, f)$ respectively. In each of the games, the distinguisher $D$ interacts with a Left oracle $L$ and a right oracle $R$. We choose $f, F$ independently and uniformly at random from all functions mapping $\{0,1\}^n$ to $\{0,1\}^n$ and run Games $G_1$ and $G_2$ using $f, F$ as follows.

- **Game** $G_1$. In Game $G_1$, on a Left oracle query $uv$ by $D$ (where $u, v \in \{0,1\}^{n/2}$), $L$ returns $F(uv)$. On a Right oracle query $uv$ by $D$ (where $u, v \in \{0,1\}^{n/2}$), $R$ returns $S^{F,f}(uv)$.
- **Game** $G_2$. In Game $G_2$, on a Left oracle query $uv$ by $D$ (where $u, v \in \{0,1\}^{n/2}$), $L$ computes $I^f(uv)$ by making two queries to $f$ and returns the result. On a Right oracle query $uv$ by $D$ (where $u, v \in \{0,1\}^{n/2}$), $R$ returns $f(uv)$.

In the discussions below, in both $G_1$ and $G_2$, we abuse the notation $L(uv)$ to refer to two things. When we say query $L(uv)$, we refer to a query $uv$ made by $D$ to the Left oracle $L$ and when $L(uv)$ is preceded or followed by an assignment or $\oplus$ operation, we refer to the the result of a query $uv$ made by $D$ to the Left oracle $L$. When $L(uv)$ is used in the latter sense, we let $L(uv) = L^1(uv)L^2(uv)$ where $|L^1(uv)| = |L^2(uv)| = n/2$. Also, the notation $L_{2j-1}(uv)$ (where $1 \leq j \leq q$) refers to the $j^{th}$ $L$ query $uv$ made by $D$ or the output of the $j^{th}$ $L$ query $uv$ made by $D$, depending on the sense in which it is used. Similarly, when we say query

$R(uv)$, we refer to a query $uv$ made by $D$ to the Right oracle $R$ and when $R(uv)$ is preceded or followed by an assignment or $\oplus$ operation, we refer to the the result of a query $uv$ made by $D$ to the Right oracle $R$. When $R(uv)$ is used in the latter sense, we let $R(uv) = R^1(uv)R^2(uv)$ where $|R^1(uv)| = |R^2(uv)| = n/2$. Also, the notation $R_{2j}(uv)$ (where $1 \leq j \leq q$) refers to the $j^{th}$ $R$ query $uv$ made by $D$ or the output of the $j^{th}$ $R$ query $uv$ made by $D$, depending on the sense in which it is used. Here $u, v \in \{0,1\}^{n/2}$. So, the query sequence made by $D$ in each game is as follows.

**Query sequence**

$$L_1(x_1y_1), R_2(x_2y_2), L_3(x_3y_3), R_4(x_4y_4), ...., L_{2q-1}(x_{2q-1}y_{2q-1}), R_{2q}(x_{2q}y_{2q})$$

for some $x_1, y_1, ..., x_{2q}, y_{2q} \in \{0,1\}^{n/2}$.

For each $i \in \{1,2\}$, let $p_i$ be the probability that $D$ accepts in Game $G_i$ (The probability $p_1$ is over the choice of $f, F$ and the probability $p_2$ is over the choice of $f$). We would like to show to show that $|p_2 - p_1|$ is close to 0. To this extent, we take the following approach.

Following Bernstein [Ber05], we would like to fix a query-answer sequence of length $2q$, corresponding to the $2q$ queries ($q$ queries to $L$ and $q$ queries to $R$ and the queries are in the form of an $L$ query, an $R$ query, $L$ query, an $R$ query and so on) and say that for each such fixed query-answer sequence, the probability (over the choice of $f$) that we get that answer sequence for the given query sequence in Game 2 is greater than or equal to $(1-\epsilon)$ times the probability (over the choice of $f, F$) that we get that answer sequence for the given query sequence in Game 1. Then we can use that to prove that $|p_1 - p_2|$ is close to 0 in a straightforward way by using Theorem 3.1 of Bernstein [Ber05].

However, this is clearly not true for all such fixed query-answer sequences of length $2q$, because there exists some query-answer sequences that occur with zero probability in Game 2, but occur with a positive probability in Game 1. For example, consider the following event. Suppose there exists queries $L_i(uv)$ and $R_j(u0^{n/2})$ (for some $u, v \in \{0,1\}^{n/2}$ and for some $i, j$, $1 \leq i \neq j \leq 2q$) made by $D$, such that $R_j(u0^{n/2}) = (u \oplus v)0^{n/2}$ and $L_i(uv) \neq (u \oplus v)0^{n/2}$. It is clear that in Game $G_2$, this event occurs with zero probability. This is because of the definition of $L$ and $R$ in Game $G_2$: if $R_j(u0^{n/2}) = (u \oplus v)0^{n/2}$, then $L_i(uv) = (u \oplus v)0^{n/2}$. However, this is not the case in Game $G_1$. If $L_i(uv) \neq (u \oplus v)0^{n/2}$, then $R_j(u0^{n/2})$ can still be equal to $(u \oplus v)0^{n/2}$ (although with some small probability), because of the definition of $L$ and $R$ in Game $G_1$. So, for a fixed query-answer sequence of length $2q$ containing a query-answer sequence of this form, it is clear that the probability (over the choice of $f$) that we get that answer sequence for the given query sequence in Game 2 is less than $(1-\epsilon)$ times the probability (over the choice of $f, F$) that we get that answer sequence for the given query sequence in Game 1.

We want to rule out such query-answer sequences. For convenience, we will also rule out the query-answer sequences that happen with zero probability in both games and some query-answer sequences that happen with a positive probability in both games, and show that the probability (over the choice of $f, F$) that $D$ causes a ruled out query-answer sequences to occur in Game $G_1$ is small, say $\epsilon_1$. Then, we can proceed to show that, for all fixed query-answer sequences of length $2q$ that have not been not ruled out, the probability (over the choice of $f$) that we get that answer sequence for the given query sequence in Game 2 is greater than or equal to $(1-\epsilon)$ times the probability (over the choice of $f, F$) that we get that answer sequence for the given query sequence in Game 1[5]. This, combined with the proof that the probability (over the choice of $f, F$) that $D$ causes a ruled out query-answer sequences to occur in Game $G_1$ is small, can be used in a

[5] However it is interesting to note that, the probability (over the choice of $f$) that we get that answer sequence for the given query sequence in Game 2 will not be too much greater than the probability (over the choice of $f, F$) that we get that answer sequence for the given query sequence in Game 1. That is, we will have that the ratio of the probability (over the choice of $f$) that we get that answer sequence for the given query sequence in Game 2, to the probability (over the choice of $f, F$) that we get that answer sequence for the given query sequence in Game 1, will be close to 1. However, we would not be showing this explicitly as it is not required to do so in the proof.

to prove that $|p_2 - p_1| \leq \epsilon + \epsilon_1$ by extending Theorem 3.1 of Bernstein [Ber05] in a natural way (which we will see shortly). Summarizing the approach,

1. Define a notion of "ruled out query-answer sequences". This notion will include query-answer sequences that occur with zero probability in Game $G_2$ but with some positive probability in Game $G_1$, query-answer sequences that happen with zero probability in both games, and some query-answer sequences that happen with a positive probability in both games.

2. Prove that the probability (over the choice of $f, F$) that $D$ causes a ruled out query-answer sequence to occur in Game $G_1$ is small, say $\epsilon_1$.

3. Prove that for all fixed query-answer sequences (of length $2q$) that have not been ruled out, the probability (over the choice of $f$) that we get that answer sequence for the given query sequence in Game 2 is greater than or equal to $(1-\epsilon)$ times the probability (over the choice of $f, F$) that we get that answer sequence for the given query sequence in Game 1.

4. Use (2) and (3) to prove that $|p_2 - p_1| \leq \epsilon + \epsilon_1$.

Given 2,3, here is how one would prove 4. Note that the following lemma is a natural extension of Theorem 3.1 of Bernstein [Ber05].

**Lemma 7.3.** *Let $A$ be the set of query-answer sequences of length $2q$ (corresponding to $q$ queries to $L$ and $q$ queries to $R$) such that for each query-answer sequence $a \in A$, the queries in $a$ are what $D$ would make in response to the answers in $a$. Let $A = A_1 \cup A_2$ where $A_1$ is the set of query-answer sequences that have not been ruled out and $A_2$ is the set of query-answer sequences that have been ruled out. Assume that the following two equations (that correspond to Points 2,3 above) hold,*

$$\sum_{a \in A_2} Pr[D\, causes\, a\, in\, G_1] \leq \epsilon_1 \tag{1}$$

*For every $a \in A_1$,*

$$Pr[Queries\ in\, a\, yields\ answers\ in\, a\, in\, G_2] \geq (1-\epsilon) Pr[Queries\ in\, a\, yields\ answers\ in\, a\, in\, G_1] \tag{2}$$

*Then, we have that*

$$|p_2 - p_1| \leq \epsilon + \epsilon_1 \tag{3}$$

*Proof.* Given Equations 1 and 2, we have to prove Equation 3. We do this as follows.

Let $ACC$ be the set of query-answer sequences of length $2q$ (corresponding to $q$ queries to $L$ and $q$ queries to $R$) such that for each query-answer sequence $a \in ACC$, the queries in $a$ are what $D$ would make in response to the answers in $a$ and $D$ outputs 1 (we have that $ACC \subseteq A$). Let $ACC = ACC_1 \cup ACC_2$ where $ACC_1 = A_1 \cap ACC$, is the set of query-answer sequences from $ACC$ that have not been ruled out, and $ACC_2 = A_2 \cap ACC$, is the set of query-answer sequences from $ACC$ that have been ruled out.

We have that

$$Pr[D^{G_1} = 1] = \sum_{a \in ACC_1} Pr[D\, \text{causes}\, a\, \text{in}\, G_1] + \sum_{a \in ACC_2} Pr[D\, \text{causes}\, a\, \text{in}\, G_1] \tag{4}$$

Similarly, we have that

$$Pr[D^{G_2} = 1] = \sum_{a \in ACC_1} Pr[D\, \text{causes}\, a\, \text{in}\, G_2] + \sum_{a \in ACC_2} Pr[D\, \text{causes}\, a\, \text{in}\, G_2] \tag{5}$$

Since $ACC_2 \subseteq A_2$, Equation 1 implies that

$$\sum_{a \in ACC_2} Pr[D \,\text{causes}\, a \,\text{in}\, G_1] \leq \epsilon_1 \tag{6}$$

And since $ACC_1 \subseteq A_1$, Equation 2 implies that
For every $a \in ACC_1$,

$$Pr[\text{Queries in}\, a \,\text{yields answers in}\, a \,\text{in}\, G_2] \geq (1-\epsilon) Pr[\text{Queries in}\, a \,\text{yields answers in}\, a \,\text{in}\, G_1] \tag{7}$$

Applying Equation 6 to Equation 4, we get

$$Pr[D^{G_1} = 1] \leq \sum_{a \in ACC_1} Pr[D \,\text{causes}\, a \,\text{in}\, G_1] + \epsilon_1 \tag{8}$$

From Equation 5, we have that

$$Pr[D^{G_2} = 1] \geq \sum_{a \in ACC_1} Pr[D \,\text{causes}\, a \,\text{in}\, G_2] \tag{9}$$

Since for every $a \in ACC_1 \subseteq A_1 \subseteq A$, the answers in $a$ determine the queries in $a$ that $D$ makes, we can remove $D$ from the right hand side of the equations, and hence rewrite Equations 8 and 9 as,

$$Pr[D^{G_1} = 1] \leq \sum_{a \in ACC_1} Pr[\text{Queries in}\, a \,\text{yields answers in}\, a \,\text{in}\, G_1] + \epsilon_1 \tag{10}$$

$$Pr[D^{G_2} = 1] \geq \sum_{a \in ACC_1} Pr[\text{Queries in}\, a \,\text{yields answers in}\, a \,\text{in}\, G_2] \tag{11}$$

Using Equation 7 and summing over all query-answer sequences in $ACC_1$, we get

$$\sum_{a \in ACC_1} Pr[\text{Queries in}\, a \,\text{yields answers in}\, a \,\text{in}\, G_2] \geq (1-\epsilon) \sum_{a \in ACC_1} Pr[\text{Queries in}\, a \,\text{yields answers in}\, a \,\text{in}\, G_1] \tag{12}$$

Applying Equation 12 in Equation 11, we get that

$$Pr[D^{G_2} = 1] \geq (1-\epsilon) \sum_{a \in ACC_1} Pr[\text{Queries in}\, a \,\text{yields answers in}\, a \,\text{in}\, G_1] \tag{13}$$

We can rewrite Equation 10 as

$$\sum_{a \in ACC_1} Pr[\text{Queries in}\, a \,\text{yields answers in}\, a \,\text{in}\, G_1] \geq Pr[D^{G_1} = 1] - \epsilon_1 \tag{14}$$

From Equation 13 and Equation 14 we get,

$$Pr[D^{G_2} = 1] \geq (1-\epsilon)(Pr[D^{G_1} = 1] - \epsilon_1) \tag{15}$$

Simplifying Equation 15 we get,

$$Pr[D^{G_2} = 1] - Pr[D^{G_1} = 1] \geq -\epsilon - \epsilon_1 + \epsilon\epsilon_1 \tag{16}$$

So we get,

$$p_2 - p_1 \geq -\epsilon - \epsilon_1 + \epsilon\epsilon_1 \tag{17}$$

And hence,

$$p_1 - p_2 \leq \epsilon + \epsilon_1 \tag{18}$$

We can now use a similar argument to get

$$Pr[D^{G_2} = 0] - Pr[D^{G_1} = 0] \geq -\epsilon - \epsilon_1 + \epsilon\epsilon_1$$

And hence,

$$p_2 - p_1 \leq \epsilon + \epsilon_1 \tag{19}$$

From Equation 18 and Equation 19, we get

$$|p_2 - p_1| \leq \epsilon + \epsilon_1 \tag{20}$$

□

**Step 1**

We want to rule out query-answer sequences that happen with positive probability in Game $G_1$, but with zero probability in Game $G_2$. As stated earlier, for convenience, we will also rule out sequences that happen with zero probability in both games and some extra query-answer sequences that occur with positive probability in both games. Let us call the query-answer sequences that we wish to rule out as *Bad* query-answer sequences. It will turn out the these Bad query answer-sequences collectively happen with small probability in Game $G_1$. By this we mean that, it will be unlikely that $D$ will cause a Bad query-answer sequence in Game $G_1$. Formally, we define a Bad query-answer sequence as follows.

**Definition 7.4** (Bad query-answer sequence)**.** *A query-answer sequence of length* $2q$ *(corresponding to* $q$ *queries to* $L$ *and* $q$ *queries to* $R$*) is called a Bad query-answer sequence, if it contains atleast one of the following query-answer sequences.*

1. *We have queries* $R_i(uv), R_j(st)$ *for some* $u, v, s, t \in \{0,1\}^{n/2}$ *and for some* $i, j$, $1 \leq i, j \leq 2q$ *s.t.*
   $R_j^2(st) = 0^{n/2}$ *or*
   $i \neq j$ *and* $R_j^2(st) = R_i^2(uv)$.

   *In other words, the right half of the result of the query* $R_j(st)$ *(for some* $s, t \in \{0,1\}^{n/2}$*) is equal to* $0^{n/2}$ *or the right half of the result of a different* $R$ *query.*

2. *We have queries* $L_i(uv), R_j(u0^{n/2}), R_k((\alpha\beta) \oplus (v0^{n/2}))$ *for some* $u, v, \alpha, \beta \in \{0,1\}^{n/2}$ *and for some distinct* $i, j, k$, $1 \leq i, j, k \leq 2q$ *s.t.*
   $R_j(u0^{n/2}) = \alpha\beta$ *and*
   $L_i(uv) \neq R_k((\alpha\beta) \oplus (v0^{n/2}))$

   *In other words, this query-answer sequence is essentially an inconsistency between the results of an* $L$ *query and two corresponding* $R$ *queries.*

We say that event $B_p$ has happened in Game $G_q$ to denote that above query-answer sequence property $p$ (where $p \in \{1, 2\}$) has happened in Game $G_q$ (where $q \in \{1, 2\}$).

Now, we proceed to prove that it is unlikely that $D$ would cause $B_1$ or $B_2$ in $G_1$.

**Step 2**

In this step, we prove that it is unlikely that $D$ would cause a Bad query-answer sequence in $G_1$.

Note that in Game $G_1$, $L$ never queries $f$ and a query to $S^{F,f}$ is only made on a query to $R$ by $D$ by definition of $G_1$. So, the queries to $S^{F,f}$ are all distinct since $D$ does not repeat a query to $R$. Also note that in $G_1$, $S^{F,f}$ on a new query, queries either $F$ (line 9 of Algorithm 1) or $f$ (line 13 of Algorithm 1). We first prove the following lemma about Game $G_1$.

**Lemma 7.5.** *In Game $G_1$, every time $S^{F,f}$ makes a query to $f$, it is a new query to $f$, and every time it makes a query to $F$, it is a new query to $F$.*

*Proof.* Suppose on a new query $uv$ (where $u, v \in \{0,1\}^{n/2}$) to $S^{F,f}$, it makes a query to $f$. The query it makes to $f$ is $uv$, by line 13 of Algorithm 1. This is a new query to $f$ because $uv$ is a new query to $S^{F,f}$. We now prove that if on a new query to $S^{F,f}$, it makes a query to $F$, then it is a new query to $F$. Suppose that there exists two queries $uv$ and $u'v'$ made to $S^{F,f}$ such that, $uv \neq u'v'$ and both make the simulator query $F$, however the corresponding $F$ queries made by the simulator are identical. By lines 8-9 of Algorithm 1, the queries to $F$ must be $(a(b \oplus u))$ and $(a(b' \oplus u'))$ respectively, such that $(b \oplus u) = (b' \oplus u')$ for some $a, b, b' \in \{0,1\}^{n/2}$. By line 7 of Algorithm 1, this must mean that $(a0^{n/2}, bv), (a0^{n/2}, b'v') \in H^S$ and $bv \neq b'v'$ for some $a, b, b' \in \{0,1\}^{n/2}$. However, this is not possible since $H^S$ is a partial function. Hence, $uv$ must be equal to $u'v'$. □

In the following lemma, we show that the probability that $D$ causes the event $B_1$ to occur in $G_1$ is small.

**Lemma 7.6.** *We have that, $Pr[D\ causes\ B_1\ to\ occur\ in\ G_1] \leq \frac{q(q+1)}{2.2^{n/2}}$ (where the probability is over the choice of $f, F$). Here $D$ makes exactly $q$ queries to the $L$ and exactly $q$ queries to $R$.*

*Proof.* In $G_1$, we know by Lemma 7.5 that every time $S^{F,f}$ makes a query to $f$, it is a new query to $f$ (and hence returns a new random answer) and every time it makes a query to $F$, it is a new query to $F$ (and hence returns a new random answer). Note that $D$ makes exactly $q$ queries to the $R$ and hence, exactly $q$ queries are made to $S^{F,f}$. Hence, it is easy to see that the probability that $D$ causes $B_1$ to occur in Game $G_1$ is at most $\frac{q+\binom{q}{2}}{2^{n/2}}$ . Hence, we have that $Pr[D\,\text{causes}\,B_1\,\text{to occur in}\,G_1] \leq \frac{1}{2}\frac{q(q+1)}{2^{n/2}}$. □

To show that it is unlikely that $D$ causes the event $B_2$ to occur in $G_1$, we first prove the following lemma.

**Lemma 7.7.** *Assume that $B_1$ does not occur in $G_1$. Then in Game $G_1$, for all $u, v \in \{0,1\}^{n/2}$ and distinct $i, j, k$, $1 \leq i, j, k \leq 2q$ s.t. $k > j$, the queries $L_i(uv)$, $R_j(u0^{n/2})$ and $R_k(R_j(u0^{n/2}) \oplus (v0^{n/2}))$ have been made by the $D$, we have that,*

$$L_i(uv) = R_k(R_j(u0^{n/2}) \oplus (v0^{n/2}))$$

*Proof.* Suppose that there exists $u, v \in \{0,1\}^{n/2}$ and distinct $i, j, k$, $1 \leq i, j, k \leq 2q$ where $k > j$, such that the queries $L_i(uv)$, $R_j(u0^{n/2})$ and $R_k(R_j(u0^{n/2}) \oplus (v0^{n/2}))$ have been made by $D$ and $L_i(uv) \neq R_k(R_j(u0^{n/2}) \oplus (v0^{n/2}))$.

Let $t_1t_2 = R_j(u0^{n/2})$ (this will be $f(u0^{n/2})$ due to the absence of $B_1$), where $t_1, t_2 \in \{0,1\}^{n/2}$. Now, when the query $R_k((t_1t_2) \oplus (v0^{n/2}))$ is made (that is, the query $R_k(R_j(u0^{n/2}) \oplus (v0^{n/2}))$ is made) by $D$, then according to lines 7-10 of the simulator, the simulator will reply with $F(u(t_1 \oplus t_1 \oplus v))$ (which is or will be equal to $L_i(uv)$), since $(u0^{n/2}, t_1t_2) \in H^S$ and the absence of event $B_1$ ensures that the pair $(u, t_1)$ is unique. So $L_i(uv)$ will be equal to $R_k(R_j(u0^{n/2}) \oplus (v0^{n/2}))$. Note that when $j = 1$, the equality in this lemma holds irrespective of whether $B_1$ occurs or not (because of the definition of the simulator).

□

Now we use Lemma 7.7 to prove that it is unlikely that $D$ would cause $B_2$ to occur in $G_1$.

**Lemma 7.8.** *We have that,*

1. *$Pr[D\ causes\ B_2\ with\ k > j\ to\ occur\ in\ G_1] \leq Pr[D\ causes\ B_1\ to\ occur\ in\ G_1]$*
2. *$Pr[D\ causes\ B_2\ with\ k < j\ to\ occur\ in\ G_1] \leq \frac{q}{2^n}$*

*where the probability is over the choice of $f, F$. Here $D$ makes exactly $q$ queries to the $L$ and exactly $q$ queries to $R$.*

*Proof.* We prove the cases separately.

1. $k > j$. In this case, by Lemma 7.7, we know that
   $Pr[D \,\text{causes}\, B_2 \,\text{with}\, k > j \,\text{to occur in}\, G_1 \mid B_1 \,\text{does not occurs in}\, G_1] = 0$.
   Hence, we have that $Pr[D \,\text{causes}\, B_2 \,\text{with}\, k > j \,\text{to occur in}\, G_1] \leq Pr[D \,\text{causes}\, B_1 \,\text{to occur in}\, G_1]$.

2. $k < j$. In $G_1$, we know by Lemma 7.5 that every time $S^{F,f}$ makes a query to $f$, it is a new query to $f$ (and hence returns a new random answer) and every time it makes a query to $F$, it is a new query to $F$ (and hence returns a new random answer). Note that $D$ makes exactly $q$ queries to the $R$ and hence exactly $q$ queries are made to $S^{F,f}$. For $B_2$ (with $k < j$) to occur, $D$ must correctly guess the result of a query $R(u0^{n/2})$ (the value $\alpha\beta$) before it has even made the query (note that this means the partial function $H^S$ has not yet been defined on $(u0^{n/2})$ because only $R$ makes the queries to $S^{F,f}$ in $G_1$) and this can be done with probability $\frac{1}{2^n}$. Hence, we have that $Pr[D \,\text{causes}\, B_2 \,\text{with}\, k < j \,\text{to occur in}\, G_1] \leq \frac{q}{2^n}$.

□

Now we use Lemmas 7.6 and 7.8 to conclude that it is unlikely that $D$ would cause a Bad query-answer sequence in $G_1$.

**Lemma 7.9.** *Let $B$ be the event that $D$ causes $B_1$ or $B_2$ to occur in $G_1$ (that is, $B$ is the event that $D$ causes a Bad query-answer sequence to occur in $G_1$). Then we have that, $Pr[B] \leq \frac{1}{2}\frac{q(q+1)}{2^{n/2}} + \frac{q}{2^n}$ (where the probability is over the choice of $f, F$). Here $D$ makes exactly $q$ queries to the $L$ and exactly $q$ queries to $R$.*

*Proof.* By Lemma 7.8, $Pr[D \,\text{causes}\, B_2 \,\text{with}\, k > j \,\text{to occur in}\, G_1] \leq Pr[D \,\text{causes}\, B_1 \,\text{to occur in}\, G_1]$. Hence, we have that $Pr[B] \leq Pr[D \,\text{causes}\, B_1 \,\text{to occur in}\, G_1] + Pr[D \,\text{causes}\, B_2 \,\text{with}\, k < j \,\text{to occur in}\, G_1]$. By Lemma 7.6, we have that $Pr[D \,\text{causes}\, B_1 \,\text{to occur in}\, G_1] \leq \frac{1}{2}\frac{q(q+1)}{2^{n/2}}$. By Lemma 7.8, we have that $Pr[D \,\text{causes}\, B_2 \,\text{with}\, k < j \,\text{to occur in}\, G_1] \leq \frac{q}{2^n}$. Hence, we have that $Pr[B] \leq \frac{1}{2}\frac{q(q+1)}{2^{n/2}} + \frac{q}{2^n}$. □

This completes Step 2.

## Step 3

In this step, we prove that for all fixed query-answer sequences (of length $2q$) that are not Bad, the probability (over the choice of $f$) that we get that answer sequence for the given query sequence in Game 2 is greater than or equal to $(1-\epsilon)$ times the probability (over the choice of $f, F$) that we get that answer sequence for the given query sequence in Game 1.

**Definition 7.10** (Valid query-answer sequence)**.** *We call a fixed query-answer sequence of length $2q$ (corresponding to $q$ queries to $L$ and $q$ queries to $R$) to be a Valid query-answer sequence if it is not Bad.*

Let the following be a fixed Valid query-answer sequence of length $2q$.

**Fixed Valid Query-Answer sequence**

$$L(x_1y_1) = m_1n_1, R(x_2y_2) = m_2n_2, ......, L(x_{2q-1}y_{2q-1}) = m_{2q-1}n_{2q-1}, R(x_{2q}y_{2q}) = m_{2q}n_{2q}$$

So, for $1 \leq i \leq 2q$, when $i$ is odd, $x_iy_i$ represents a query to the Left oracle and $m_in_i$ represents the corresponding response, and when $i$ is even, $x_iy_i$ represents a query to the Right oracle and $m_in_i$ represents the corresponding response. We know that for all odd $i, j$, $1 \leq i \neq j \leq 2q$, we have that $x_iy_i \neq x_jy_j$ and for all even $i, j$, $1 \leq i \neq j \leq 2q$, we have that $x_iy_i \neq x_jy_j$.

Now, we prove the following lemma which would allow us to only consider Valid query-answer sequences of a certain form.

**Lemma 7.11.** *Let $a$ be a fixed query-answer sequence of length $2q$. Let $a'$ be the query-answer sequence obtained by rearranging $a$ in the following way: all the Left oracle queries and the corresponding answers are moved after the final query to the Right oracle query and the corresponding answer, and the order of Right oracle queries and the corresponding answers are unchanged (that is, all the Left oracle queries are made after the final Right oracle query is made, and the order of Right oracle queries is unchanged). We have that the probability (over the choice of $f, F$) that the query-answer sequence $a$ occurs in Game $G_1$ is same as the probability (over the choice of $f, F$) that the query-answer sequence $a'$ occurs in Game $G_1$, and the probability (over the choice of $f$) that the query-answer sequence $a$ occurs in Game $G_2$ is same as the probability (over the choice of $f$) that the query-answer sequence $a'$ occurs in Game $G_2$.*
*Formally, let the following be a fixed query-answer sequence of length $2q$*

$$L(x_1y_1) = m_1n_1, R(x_2y_2) = m_2n_2, ......, L(x_{2q-1}y_{2q-1}) = m_{2q-1}n_{2q-1}, R(x_{2q}y_{2q}) = m_{2q}n_{2q}$$

*Then we have that,*
*(1)*$Pr[L(x_1y_1) = m_1n_1, R(x_2y_2) = m_2n_2, ......, L(x_{2q-1}y_{2q-1}) = m_{2q-1}n_{2q-1}, R(x_{2q}y_{2q}) = m_{2q}n_{2q} \, in \, G_1]$
$= Pr[R(x_2y_2) = m_2n_2, ..., R(x_{2q}y_{2q}) = m_{2q}n_{2q}, L(x_1y_1) = m_1n_1, ..., L(x_{2q-1}y_{2q-1}) = m_{2q-1}n_{2q-1} \, in \, G_1]$
*(2)*$Pr[L(x_1y_1) = m_1n_1, R(x_2y_2) = m_2n_2, ......, L(x_{2q-1}y_{2q-1}) = m_{2q-1}n_{2q-1}, R(x_{2q}y_{2q}) = m_{2q}n_{2q} \, in \, G_2]$
$= Pr[R(x_2y_2) = m_2n_2, ..., R(x_{2q}y_{2q}) = m_{2q}n_{2q}, L(x_1y_1) = m_1n_1, ..., L(x_{2q-1}y_{2q-1}) = m_{2q-1}n_{2q-1} \, in \, G_2]$

*Proof.* We prove each case separately.

1. In Game $G_1$, on a query $uv$ to $L$, $L$ returns $F(uv)$. So, it does not matter in which order the $L$ queries are made. However, on a query $uv$ to $R$, $R$ returns $S^{F,f}(uv)$. The simulator queries either $F$ or $f$ based on its history $H^S$. In other words, the decision made by the deterministic simulator is entirely dependant on the queries it has seen and the responses it has made (also note that the simulator does not see the queries made to $L$). So, the order in which the $R$ queries are made do matter. Hence, as long as the order in which the $R$ queries are made is maintained, it does not matter in which order the rest of the queries (that is, the $L$ queries) are made. Hence, we get identical answers if we move all the $L$ queries after the final $R$ query.

2. In Game $G_2$, on a query $uv$ to $L$, $L$ computes $I^f(uv)$ by making two calls to $f$ and returns the result. On a query $uv$ to $R$, $R$ returns $f(uv)$. So it does not matter in which order the queries to $L$ and $R$ made. Hence, we get identical answers if we move all the $L$ queries after the final $R$ query.

□

It is important to note that if we take a Valid query-answer sequence and rearrange them such that it is of the above form (all the $L$ queries are moved after the last $R$ query), the resultant query-answer sequence is still Valid. This is because of the following reason. If we examine the definition of Bad query-answer sequence (Definition 7.4), neither of the bad query-answer sequence properties depend on the order in which the queries are made. Hence the above rearrangement of a Valid query-answer sequence cannot lead to a Bad query-answer sequence.

Let the following be a fixed Valid query-answer sequence of length $2q$ (due to Lemma 7.11, it is sufficient to consider the query-answer sequences of such form),

**Fixed Valid Query-Answer sequence**

$$R(x_1y_1) = m_1n_1, ..., R(x_qy_q) = m_qn_q, L(x_{q+1}y_{q+1}) = m_{q+1}n_{q+1}, ..., L(x_{2q}y_{2q}) = m_{2q}n_{2q}$$

Note the minor change in notation here. When $1 \leq i \leq q$, $x_iy_i$ represents a query to the Right oracle and $m_in_i$ represents the corresponding response and when $q+1 \leq i \leq 2q$, $x_iy_i$ represents a query to the Left oracle and $m_in_i$ represents the corresponding response. This is purely for the sake of convenience.

In each game, we want to calculate the probability that $R(x_1y_1) = m_1n_1, ..., R(x_qy_q) = m_qn_q, L(x_{q+1}y_{q+1}) = m_{q+1}n_{q+1}, ..., L(x_{2q}y_{2q}) = m_{2q}n_{2q}$. That is, we want to calculate the probability that $2q$ events (for each of the $2q$ query-answer pairs, the event that the query results in the answer) happen in the given order. This probability is the product of the probabilities that an event occurs given that previous events have occurred. Since we are considering only Valid query-answer sequences, in each game, it will turn out that none of these $2q$ events (given that previous events have occurred) happen with zero probability. However, it is possible that there are some query-answer pairs that are completely determined by previous query-answer pairs in both the games and hence these query-answer pairs (that are determined) would occur with probability 1 in both the games. We will see that the rest of the events (given that previous events have occurred) occur with probability $1/2^n$ in Game $G_1$ and close to $1/2^n$ in Game $G_2$. Now, we define type $E$ query-answer pairs, which intuitively are query-answer pairs that are determined by previous query-answer pairs. The idea is that, a type $E$ query-answer pair is an $L$ query-answer pair which is determined from two previous $R$ query-answer pairs in the obvious way. To be more precise,

**Definition 7.12** (A Type $E$ query-answer pair). *A query-answer pair $L(x_iy_i) = m_in_i$ (for some $i$, $q+1 \leq i \leq 2q$) is of Type E if for some $j, k$ s.t. $1 \leq j \neq k \leq q$, there exists query-answer pairs of the form*

- *$R(x_jy_j) = m_jn_j$ where $x_jy_j = x_i0^{n/2}$ and*
- *$R(x_ky_k) = m_kn_k$ where $x_ky_k = (m_jn_j) \oplus y_i0^{n/2}$*

For a query-answer pair $L(x_iy_i) = m_in_i$ of type $E$, it would follow that $m_kn_k = m_in_i$ in both the games (because the query-answer sequences are Valid). Hence, the probability that the outcome of $L(x_iy_i) = m_in_i$ given that the outcome of $R(x_jy_j) = m_jn_j$ and the outcome of $R(x_ky_k) = m_kn_k$, is 1. Let $p$ denote the number of query-answer pairs of type $E$ in the fixed Valid query-answer sequence. From the definition of Type $E$ query-answer pairs, it follows that $1 \leq p \leq q$.

Now, we proceed to prove the following two lemmas which calculate the probability that $R(x_1y_1) = m_1n_1, ..., R(x_qy_q) = m_qn_q, L(x_{q+1}y_{q+1}) = m_{q+1}n_{q+1}, ..., L(x_{2q}y_{2q}) = m_{2q}n_{2q}$ in each game.

For convenience, in each game, let us use that notation $T_i$ to denote the event that the outcome of the $i^{th}$ query $x_iy_i$ is $m_in_i$ (for every $i$, $1 \leq i \leq 2q$). So, when $1 \leq i \leq q$, $T_i$ denotes the event that the outcome of $R(x_iy_i) = m_in_i$ and when $q+1 \leq i \leq 2q$, $T_i$ denotes the event that the outcome of $L(x_iy_i) = m_in_i$

**Lemma 7.13.** *Let $R(x_1y_1) = m_1n_1, ..., R(x_qy_q) = m_qn_q, L(x_{q+1}y_{q+1}) = m_{q+1}n_{q+1}, ..., L(x_{2q}y_{2q}) = m_{2q}n_{2q}$ be a fixed Valid query-answer sequence. The probability (over the choice of $f, F$) that the outcome $R(x_1y_1) = m_1n_1, ..., R(x_qy_q) = m_qn_q, L(x_{q+1}y_{q+1}) = m_{q+1}n_{q+1}, ..., L(x_{2q}y_{2q}) = m_{2q}n_{2q}$ occurs in Game $G_1$ is $\frac{1}{(2^n)^{2q-p}}$.*

*Proof.* In Game $G_1$, we want the probability that $2q$ events (for each of the $2q$ query-answer pairs, the event that the query results in the answer) happen in the given order. This probability is the product of the probabilities that an event occurs given that previous events have occurred.

Formally we have,

$$\begin{aligned} &Pr[T_1 \wedge T_2 \wedge ... \wedge T_{2q} \,\text{in}\, G_1] \\ &= Pr[T_1] \;\; Pr[T_2 \mid T_1] \;\; Pr[T_3 \mid T_1 \wedge T_2] ... \\ &\quad Pr[T_{2q} \mid T_1 \wedge T_2 \wedge ... \wedge T_{2q-1}] \end{aligned} \tag{21}$$

**Right oracle query-answer pairs** For each Right oracle query-answer pair $R(x_iy_i) = m_in_i$ ($1 \leq i \leq q$), we calculate the probability that the stipulated answer to the query occurs, given that stipulated answers to previous queries have occurred. In Game $G_1$, on a query $x_ky_k$ to the Right oracle $R$ ($1 \leq i \leq q$), $R$ returns $S^{f,F}(x_iy_i)$. The simulator $S^{f,F}$ on $x_iy_i$ calls either $f$ or $F$. Now, by Lemma 7.5, we know that every time

$S^{F,f}$ makes a query to $f$, it is a new query to $f$ (and hence results in a new random answer $m_in_i$) and every time it makes a query to $F$, it is a new query to $F$ (and hence results in a new random answer $m_in_i$). Since, the $R$ queries are made before any $L$ query is made, we have that for each $i$, $1 \leq i \leq q$

$$Pr[T_i \mid T_1 \wedge ... \wedge T_{i-1}] = 1/2^n \tag{22}$$

**Left oracle query-answer pairs** For each Left oracle query-answer pair $L(x_iy_i) = m_in_i$ ($q+1 \leq i \leq 2q$), we calculate the probability that the stipulated answer to the query occurs, given that stipulated answers to previous queries have occurred.

We know that for each query-answer pair $L(x_iy_i) = m_in_i$ that is of the type $E$ (note that there can be at most $p$ query-answer pairs of type $E$),

$$Pr[T_i \mid T_1 \wedge ... \wedge T_{i-1}] = 1 \tag{23}$$

Now we consider the query-answer pairs that are not of type $E$. In Game $G_1$, on a query $x_iy_i$ to the Left oracle $L$ ($q + 1 \leq i \leq 2q$), $L$ returns $F(x_iy_i)$. It is not difficult to see that this is the first time $F$ sees the query $(x_iy_i)$ and hence the result $m_in_i$ will be a new random answer. Suppose the query $(x_iy_i)$ to $F$ at time $i$ is not a new query to $F$. That is, $x_iy_i$ had been made to $F$ at some time $k < i$. We rule out the case $q + 1 \leq k < i$ (that is, it was made as a previous $L$ query at time $k$) because the queries to $L$ are distinct. We now consider the case that $1 \leq k \leq q$ (that is, it was made as a result of a previous $R$ query at time $k$). That is, $R$ on a query $x_ky_k$, had made a query $x_iy_i$ to $F$. Then, it must mean that there exists a query-answer pair $(x_jy_j, m_jn_j)$ ($1 \leq j \leq q$) such that $x_jy_j = x_i0^{n/2}$ and $n_j = y_k$ and $m_j = x_k \oplus y_i$ (equivalently, $x_ky_k = (m_jn_j) \oplus y_i0^{n/2}$). Hence, the query-answer pair $L(x_iy_i) = m_in_i$ at time $i$, will be of type $E$. Hence, we have that for each query-answer pair $L(x_iy_i) = m_in_i$ that is not of the type $E$,

$$Pr[T_i \mid T_1 \wedge ... \wedge T_{i-1}] = 1/2^n \tag{24}$$

Applying Equations 22, 23 and 24 in Equation 21, we get that,

$$Pr[T_1 \wedge T_2 \wedge ... \wedge T_{2q} \,\text{in}\, G_1] = \frac{1}{(2^n)^{2q-p}}$$

□

**Lemma 7.14.** *Let $R(x_1y_1) = m_1n_1, ..., R(x_qy_q) = m_qn_q, L(x_{q+1}y_{q+1}) = m_{q+1}n_{q+1}, ..., L(x_{2q}y_{2q}) = m_{2q}n_{2q}$ be a fixed Valid query-answer sequence. The probability (over the choice of $f$) that the outcome $R(x_1y_1) = m_1n_1, ..., R(x_qy_q) = m_qn_q, L(x_{q+1}y_{q+1}) = m_{q+1}n_{q+1}, ..., L(x_{2q}y_{2q}) = m_{2q}n_{2q}$ occurs in Game $G_2$ is at least $\frac{1-\epsilon}{(2^n)^{2q-p}}$ where $\epsilon = \frac{3q^2}{2^{n/2}}$.*

*Proof.* In Game $G_2$, we want the probability that $2q$ events (for each of the $2q$ query-answer pairs, the event that the query results in the answer) happen in the given order. This probability is the product of the probabilities that an event occurs given that previous events have occurred.

Formally we have,

$$\begin{aligned} &Pr[T_1 \wedge T_2 \wedge ... \wedge T_{2q} \,\text{in}\, G_2] \\ &= Pr[T_1]\ Pr[T_2 \mid T_1]\ Pr[T_3 \mid T_1 \wedge T_2] ... \\ &\quad Pr[T_{2q} \mid T_1 \wedge T_2 \wedge ... \wedge T_{2q-1}] \end{aligned} \tag{25}$$

We now introduce certain events that are specific to Game $G_2$, which we call the $f-$events. Intuitively, $f-$ events are unlikely coincidences that can occur as a result of a value being assigned to a first call to $f$ (due to an $L$ query) that has not been made before, which can potentially cause the answer to the

$L$ query to not look random (Shortly, we will see that the answers to Right oracle queries are always random).

Here, we abuse the notation $f(uv)$ to refer to two things. When we say query $f(uv)$, we refer to a query $uv$ made by to $f$ (by either $R$ or $L$) and when $f(uv)$ is preceded or followed by an assignment or $\oplus$ operation, we refer to the the result of a query $uv$ made to $f$ (by $L$ or $R$). When $f(uv)$ is used in the latter sense, we let $f(uv) = f^1(uv)f^2(uv)$ where $|f^1(uv)| = |f^2(uv)| = n/2$ $(u, v \in \{0,1\}^{n/2})$.

Formally,

**Definition 7.15** ($f-$event)**.** *We define an $f-$ event happening at time $i$, where $1 \leq i \leq 2q$ as follows*

**Case (a)** $1 \leq i \leq q$ *(That is, $i$ represents an $R$ query made by the adversary)*
*Let $\alpha = f^2(x_iy_i)$. We say an $f-$ event happens at time $i$ if at least one of the following holds*

1. $\alpha = 0^{n/2}$
2. $\alpha = f^2(x_jy_j)$ *for some $j$, $1 \leq j < i$*

*Note that since the query-answer sequence is Valid, for every $i$, $1 \leq i \leq q$, it will **not** be the case that $\alpha = 0^{n/2}$ or $\alpha = f^2(x_jy_j)$ for some $j$, $1 \leq j < i$. However, we are including this as part of $f-$ events for the sake of better exposition.*

**Case (b)** $q+1 \leq i \leq 2q$ *(That is, $i$ represents an $L$ query made by the adversary)*
*Let $\alpha = f^2(x_i0^{n/2})$. We say an $f-$ event happens at time $i$ if*

- *$x_j \neq x_i$ for all $j$, $q+1 \leq j < i$ and $x_jy_j \neq x_i0^{n/2}$ for all $j$, $1 \leq j \leq q$ (that is, for an $f-$ event to happen at time $i$, the query $f(x_i0^{n/2})$ must not have been made before due to an $R$ query or as a first call due to an $L$ query at some time $j < i$) and*
  *At least one of the following holds*
  1. $\alpha = 0^{n/2}$ *or*
  2. $\alpha = y_j$ *for some $j$, $1 \leq j \leq q$ or*
  3. $\alpha = f^2(x_j0^{n/2})$ *for some $j$, $q+1 \leq j < i$ or*
  4. $\alpha = f^2(x_jy_j)$ *for some $j$, $1 \leq j \leq q$*

**Remark 7.16.** *Note that, for every $L$ query-answer pair, $L(x_iy_i) = m_in_i$, the following holds. If $f^2(x_i0^{n/2})$ at time $i$ satisfies at least one of the following of Definition 7.15,*

- *The equality mentioned in case $(b) : 1$ or*
- *The equality mentioned in case $(b) : 3$, with the constraint that $x_i \neq x_j$ or*
- *The equality mentioned in case $(b) : 4$, with the constraint that $x_i0^{n/2} \neq x_jy_j$,*

*Then it means that an $f-$ event occurs at some time $t$, $1 \leq t \leq i$.*

For convenience, let us use the notation $S_i$ (where $1 \leq i \leq 2q$) to denote the event that no $f-$ event has happened before time $i$.

The outline for the key part of the proof of this lemma that deals with the $L$ query-answer pairs that are not of the type $E$ is as follows-

For each query-answer pair $L(x_iy_i) = m_in_i$ that is not of the type $E$, given that

- No $f-$ events have occurred before time $i$ (that is, $S_i$ holds) and
- Stipulated answers to previous queries have occurred (that is, $T_1 \wedge T_2 \wedge ... \wedge T_{i-1}$ holds).

Then we will see that,

1. With very high probability, no $f-$ event occurs at time $i$.
2. Given 1, the second call made by $L$ to $f$ at time $i$ is a new call to $f$.
3. Given 2, the probability that the outcome of $L(x_iy_i) = m_in_i$ is $1/2^n$.

We will see that 3 is a straightforward argument. Also, proving 1 will be quite easy. However, proving 2 is the non-trivial part and will contribute to bulk of this proof.

Now, we can rewrite Equation 25 as,

$$\begin{aligned}
&Pr[T_1 \wedge T_2 \wedge ... \wedge T_{2q} \,\text{in}\, G_2] \\
&\geq Pr[T_1 \wedge T_2 \wedge ... \wedge T_{2q} \,\text{and no}\, f - \text{ event occurs} \,\text{in}\, G_2] \\
&= Pr[T_1 \,\text{and no}\, f - \text{ event occurs at time 1}] \\
&\quad Pr[T_2 \,\text{and no}\, f - \text{ event occurs at time 2} \mid T_1 \wedge S_2] ... \\
&\quad Pr[T_{2q} \,\text{and no}\, f - \text{ event occurs at time}\, 2q \mid T_1 \wedge T_2 \wedge ... \wedge T_{2q-1} \wedge S_{2q}]
\end{aligned} \tag{26}$$

**Right oracle query-answer pairs** For each Right oracle query-answer pair $R(x_iy_i) = m_in_i$ $(1 \leq i \leq q)$, we calculate the probability that the stipulated answer to the query occurs and no $f$ event occurs at time $i$, given that stipulated answers to previous queries have occurred and no $f$ event has happened before time $i$. In Game $G_2$, on a query $x_iy_i$ to the Right oracle $R$ $(1 \leq i \leq q)$, $R$ returns $f(x_iy_i)$. Since $R$ queries are made before any $L$ query is made and the queries to $R$ are distinct, the query $x_iy_i$ is a new query to $f$ and will result in a new random answer $m_in_i$. So, irrespective of whether $f-$ event has occurred or not, the probability that the outcome of $R(x_iy_i) = m_in_i$ (for $1 \leq i \leq q$) is $1/2^n$. So, we have the following,

Since the query-answer sequence is Valid, it is easy to see that
For each $i$, $1 \leq i \leq q$,

$$Pr[\text{no}\, f - \text{ event occurs at time}\, i \mid T_1 \wedge ...T_{i-1} \wedge S_i] = 1$$

$$Pr[T_i \mid \text{no}\, f - \text{ event occurs at time } \; i \wedge T_1 \wedge ...T_{i-1} \wedge S_i] = 1/2^n$$

We have that,

$$\begin{aligned}
&Pr[T_i \,\text{and no}\, f - \text{ event occurs at time}\, i \mid T_1 \wedge ...T_{i-1} \wedge S_i] \\
&\quad = Pr[\text{no}\, f - \text{ event occurs at time}\, i \mid T_1 \wedge ...T_{i-1} \wedge S_i] \\
&\qquad Pr[T_i \mid \text{no}\, f - \text{ event occurs at time } \; i \wedge T_1 \wedge ...T_{i-1} \wedge S_i]
\end{aligned}$$

And hence, for each $i$, $1 \leq i \leq q$

$$Pr[T_i \,\text{and no}\, f - \text{ event occurs at time}\, i \mid T_1 \wedge ...T_{i-1} \wedge S_i] = 1/2^n \tag{27}$$

**Left oracle query-answer pairs** For each Left oracle query-answer pair $L(x_iy_i) = m_in_i$ $(q+1 \leq i \leq 2q)$, we calculate the probability that the stipulated answer to the query occurs and no $f$ event occurs at time $i$, given that stipulated answers to previous queries have occurred and no $f$ event has happened before time $i$. In Game $G_2$, on a Left oracle query $x_iy_i$, $L$ returns $f(f(x_i0^{n/2}) \oplus y_i0^{n/2})$ by making two calls to $f$.

For a query-answer pair $L(x_iy_i) = m_in_i$ that is of the type $E$, no $f-$ event occurs at time $i$. This is because, a type $E$ query-answer pair is completely determined by previous $R$ queries. More formally, for the query-answer pair $L(x_iy_i) = m_in_i$ that is of type $E$, we have that $x_i0^{n/2} = x_jy_j$ for some $j$, $1 \leq j \leq q$ and hence by the definition of $f-$ events, an $f-$ event cannot occur at time $i$.

Hence, for each query-answer pair $L(x_iy_i) = m_in_i$ that is of type $E$, we have that

$$Pr[\text{no}\, f - \text{ event occurs during time}\, i \mid T_1 \wedge T_2 \wedge ... \wedge T_{i-1} \wedge S_i] = 1$$

Also, since they are determined by previous $R$ queries, we know that

$$Pr[T_i \mid \text{no}\, f - \text{ event occurs during time}\, i\, \text{and}\, T_1 \wedge T_2 \wedge ... \wedge T_{i-1} \wedge S_i] = 1$$

Hence, for each query-answer pair $L(x_iy_i) = m_in_i$ that is of type $E$, we have that

$$Pr[T_i\, \text{and no}\, f - \text{ event occurs during time}\, i \mid T_1 \wedge T_2 \wedge ... \wedge T_{i-1} \wedge S_i] = 1 \tag{28}$$

Note that there are can be at most $p$ query-answer pairs that are of type $E$.

We now consider the query-answer pairs that are not of the type $E$. Following the outline described before, we prove the following claims for a query-answer pair $L(x_iy_i) = m_in_i$ (where $q+1 \leq i \leq 2q$) that is not of type $E$, assuming that $S_i \wedge T_1 \wedge T_2 \wedge ... \wedge T_{i-1}$ holds.

**Claim 7.17.** *We have that with very high probability, no $f-$ event occurs at time $i$.*

**Claim 7.18.** *Given Claim 7.17, the second call made by $L$ to $f$ at time $i$ is a new call to $f$.*

**Claim 7.19.** *Given Claim 7.18, the probability that the outcome of $L(x_iy_i) = m_in_i$ is $1/2^n$.*

**Proof of Claim 7.17** We show that given that $S_i \wedge T_1 \wedge T_2 \wedge ... \wedge T_{i-1}$ holds, then we have that with very high probability, no $f-$ event occurs at time $i$. We first show that for an $f-$ event to happen at time $i$, the first call made by $L$ to $f$ at time $i$ (that is, $f(x_i0^{n/2})$) must be a new call to $f$.

- By definition of $f-$ events, for an $f-$ event to happen at time $i$, the query $f(x_i0^{n/2})$ must not have been made before due to an $R$ query or as a first call due to an $L$ query at some time $j$ where $j < i$.
- Also, the assumption that $S_i$ holds implies that the query $f(x_i0^{n/2})$ could not have been made as a second call due to an $L$ query at some time $j$ where $j < i$. Here is the reason.
  Suppose that the first time the query $f(x_i0^{n/2})$ was made was as a second call to $f$ due to an $L$ query $L(x_jy_j)$ at some time $j < i$. So, we have that $f(x_j0^{n/2}) \oplus y_j0^{n/2} = x_i0^{n/2}$ and hence $f^2(x_j0^{n/2}) = 0^{n/2}$. This means that an $f-$ event had happened at some time $t \leq j$ (as a consequence of Remark 7.16).

So, the above two conditions imply that if an $f-$ event occurs at time $i$, then it is due to the first time the query $f(x_i0^{n/2})$ is made and that is as a first call by $L$ to $f$ at time $i$, and hence the result of the query is chosen uniformly at random. Examining Definition 7.15 (specifically cases case $(b) : 2, 3, 4, 5$), we can see that $3q + 1$ values must be ruled out and hence it is easy to see that, for each query-answer pair $L(x_iy_i) = m_in_i$ that is not of type $E$,

$$Pr[\text{no}\, f - \text{ event occurs during time}\, i \mid T_1 \wedge T_2 \wedge ... \wedge T_{i-1} \wedge S_i] \geq 1 - \frac{3q+1}{2^{n/2}}$$

□

**Proof of Claim 7.18** We show that given Claim 7.17, the second call made to $f$ (that is, $f(f(x_i0^{n/2}) \oplus y_i0^{n/2}))$ at time $i$ is a new call to $f$. We will also be using the assumption that the query-answer pair $L(x_iy_i) = m_in_i$ is not of type $E$ to show this. Suppose that the second call made to $f$ (that is, $f(f(x_i0^{n/2}) \oplus y_i0^{n/2}))$ at time $i$ is not a new call to $f$. Then, we prove that an $f-$ event had occurred at some time $t \leq i$ or the query-answer pair $L(x_iy_i) = m_in_i$ is of type $E$. We split it into three cases based on when the query $f(f(x_i0^{n/2}) \oplus y_i0^{n/2})$ was first made.

- Suppose that the first time the query $f(f(x_i0^{n/2}) \oplus y_i0^{n/2})$ was made was as a first call to $f$ due to an due an $L$ query $L(x_ky_k)$ at some time $k \leq i$. This means that $f(x_i0^{n/2}) \oplus y_i0^{n/2} = x_k0^{n/2}$ and hence $f^2(x_i0^{n/2}) = 0^{n/2}$. Hence, an $f-$ event had occurred at some time $t \leq i$ (as a consequence of Remark 7.16).

- Suppose that the first time the query $f(f(x_i0^{n/2}) \oplus y_i0^{n/2})$ was made was as a second call to $f$ due to an $L$ query $L(x_ky_k)$ at some time $k < i$. This means that $f(x_i0^{n/2}) \oplus y_i0^{n/2} = f(x_k0^{n/2}) \oplus y_k0^{n/2}$ where $x_k \neq x_i$ (if $x_k = x_i$, then we would have that $y_i = y_k$ and hence the $L$ queries at times $i$ and $k$ where $k < i$, wouldn't be distinct.) and hence $f^2(x_i0^{n/2}) = f^2(x_k0^{n/2})$ where $x_i \neq x_k$. Hence, an $f-$ event had occurred at some time $t \leq i$ (as a consequence of Remark 7.16).

- Suppose that the first time the query $f(f(x_i0^{n/2}) \oplus y_i0^{n/2})$ was made was due to an $R$ query $R(x_ky_k)$ at some time $1 \leq k \leq q$. That is, $f(x_i0^{n/2}) \oplus y_i0^{n/2} = x_ky_k$. We split this into three cases based on when the query $f(x_i0^{n/2})$ was first made.
  - Suppose that the first time the query $f(x_i0^{n/2})$ was made was as a first call to $f$ due to an $L$ query $L(x_jy_j)$ at some time $j \leq i$. Then, we have that $f^2(x_j0^{n/2}) = f^2(x_i0^{n/2})$ since $x_j = x_i$. Since we have that $f(x_i0^{n/2}) \oplus y_i0^{n/2} = x_ky_k$, we have that $f^2(x_i0^{n/2}) = y_k$ and hence $f^2(x_j0^{n/2}) = y_k$. So, we have that an $f-$ event occurred at time $j \leq i$.
  - Suppose that the first time the query $f(x_i0^{n/2})$ was made was as a second call to $f$ due to an $L$ query $L(x_jy_j)$ at some time $j < i$. Then, we have that $f(x_j0^{n/2}) \oplus y_j0^{n/2} = x_i0^{n/2}$ and hence $f^2(x_j0^{n/2}) = 0^{n/2}$. Hence, we have that an $f-$ event had occurred at some time $t \leq j$ (as a consequence of Remark 7.16).
  - Suppose that the first time the query $f(x_i0^{n/2})$ was made was due to an $R$ query $R(x_jy_j)$ at some time $1 \leq j \leq q$. That is, we have that $x_jy_j = x_i0^{n/2}$. So, it means that we have made previous $R$ queries $R(x_jy_j)$ where $x_jy_j = x_i0^{n/2}$ and $R(x_ky_k)$ where $x_ky_k = f(x_i0^{n/2}) \oplus y_i0^{n/2}$ (that is, $x_ky_k = m_jn_j \oplus y_i0^{n/2}$). Hence the query-answer pair $L(x_iy_i) = m_in_i$ will be of type $E$.

□

**Proof of Claim 7.19** We show that given Claim 7.18, the probability that the outcome of $L(x_iy_i) = m_in_i$ is $1/2^n$. This is a trivial argument because the result of the second call to $f$ (in this case, $m_in_i$) is chosen uniformly at random since the second call is a new call to $f$. So, we have that, for each query-answer pair $L(x_iy_i) = m_in_i$ that is not of the type $E$,

$$Pr[T_i \mid \text{no}\, f - \text{ event occurs during time}\, i\, \text{and}\, T_1 \wedge T_2 \wedge ... \wedge T_{i-1} \wedge S_i] = 1/2^n$$

□

Summarizing Claims 7.17, 7.18 and 7.19,

We have that,

$$\begin{aligned} &Pr[T_i\, \text{and no}\, f - \text{ event occurs at time}\, i \mid T_1 \wedge ...T_{i-1} \wedge S_i] \\ &= Pr[\text{no}\, f - \text{ event occurs at time}\, i \mid T_1 \wedge ...T_{i-1} \wedge S_i] \\ &\quad Pr[T_i \mid \text{no}\, f - \text{ event occurs at time }\ i \wedge T_1 \wedge ...T_{i-1} \wedge S_i] \end{aligned}$$

Hence, for each query-answer pair $L(x_iy_i) = m_in_i$ that is not of the type $E$,

$$Pr[T_i \,\text{and no}\, f - \,\text{event occurs at time}\, i \mid T_1 \wedge ...T_{i-1} \wedge S_i] \geq \frac{1}{2^n}(1 - \frac{3q+1}{2^{n/2}}) \tag{29}$$

We now proceed to conclude Lemma 7.14.

Applying Equations 27, 28 and 29 in Equation 26, we get that

$$\begin{aligned} &Pr[T_1 \wedge T_2 \wedge ... \wedge T_{2q} \,\text{in}\, G_2] \\ &\geq (\frac{1}{(2^n)^{2q-p}})(1 - \sum_{i=1}^{q} \frac{3q+1}{2^{n/2}}) \\ &= (\frac{1}{(2^n)^{2q-p}})(1 - \frac{3q^2+q}{2^{n/2}}) \end{aligned}$$

Hence, we have that

$$Pr[T_1 \wedge T_2 \wedge ... \wedge T_{2q} \,\text{in}\, G_2] \geq \frac{1-\epsilon}{(2^n)^{2q-p}}$$

where $\epsilon = \frac{3q^2+q}{2^{n/2}}$.

**Lemma 7.20.** *Let $L(x_1y_1) = m_1n_1, R(x_2y_2) = m_2n_2, ......, L(x_{2q-1}y_{2q-1}) = m_{2q-1}n_{2q-1}, R(x_{2q}y_{2q}) = m_{2q}n_{2q}$ be a fixed Valid query-answer sequence. Then we have that,*
*$Pr[L(x_1y_1) = m_1n_1, R(x_2y_2) = m_2n_2, ......, L(x_{2q-1}y_{2q-1}) = m_{2q-1}n_{2q-1}, R(x_{2q}y_{2q}) = m_{2q}n_{2q} \, in\, G_2]$*
*$\geq (1-\epsilon)Pr[L(x_1y_1) = m_1n_1, R(x_2y_2) = m_2n_2, ......, L(x_{2q-1}y_{2q-1}) = m_{2q-1}n_{2q-1}, R(x_{2q}y_{2q}) = m_{2q}n_{2q} \, in\, G_1]$*
*where $\epsilon = \frac{3q^2+q}{2^{n/2}}$.*

*Proof.* First apply Lemma 7.11. Then apply Lemma 7.13 and Lemma 7.14. □

**Step 4**

In this step, we will combine Steps 2 and 3 to show that $|p_1 - p_2|$ is close to 0. Recall that,

$$p_1 = Pr[D^{G_1} = 1]$$
$$p_2 = Pr[D^{G_2} = 1]$$

We use Lemmas 7.9 and 7.20 to invoke Lemma 7.3.

Let $A$ be the set of query-answer sequences of length $2q$ (corresponding to $q$ queries to $L$ and $q$ queries to $R$) such that for each query-answer sequence $a \in A$, the queries in $a$ are what $D$ would make in response to the answers in $a$. Let $A = A_1 \cup A_2$ where $A_1$ is the set of Valid query-answer sequences and $A_2$ is the set of Bad query-answer sequences.

By Lemma 7.9, we have

$$\sum_{a \in A_2} Pr[D \,\text{causes}\, a \,\text{in}\, G_1] \leq \frac{1}{2}\frac{q(q+1)}{2^{n/2}} + \frac{q}{2^n} \tag{30}$$

Lemma 7.20 implies that
$\forall a \in A_1$,

$$Pr[\text{Queries in}\, a \,\text{yields answers in}\, a \,\text{in}\, G_2] \geq (1-\epsilon)Pr[\text{Queries in}\, a \,\text{yields answers in}\, a \,\text{in}\, G_1] \tag{31}$$

where $\epsilon = \dfrac{3q^2+q}{2^{n/2}}$.

Hence, by Lemma 7.3 we have that

$$|p_2 - p_1| \leq \epsilon + \frac{1}{2}\frac{q(q+1)}{2^{n/2}} + \frac{q}{2^n} \tag{32}$$

where $\epsilon = \dfrac{3q^2+q}{2^{n/2}}$.

Hence, we conclude that

$$|p_2 - p_1| \leq \frac{1}{2}\frac{7q^2+3q}{2^{n/2}} + \frac{q}{2^n} \tag{33}$$

This completes the proof of the theorem. □

We now state the indifferentiability theorem about the construction $I^f$, for a uniform random $f$. The proof of the theorem directly follows from Theorem 7.2.

**Theorem 7.21.** *Let $X_n$ be the uniform distributions on $\mathbb{F}_n^n$, for every positive even integer $n$. Let $\mathcal{X} = \{X_n\}$. Then, construction $I$ having oracle access to the distribution family $\mathcal{X}$ is indifferentiable from the distribution family $\mathcal{X}$.*

*Proof.* Directly follows from Theorem 7.2. □

## 8 Indifferentiability Results for the Construction $I^p$, for a Uniform Random Permutation $p$

Since $I^f$, for a uniform random function $f : \{0,1\}^n \to \{0,1\}^n$, is indistinguishable from a uniform random function $F : \{0,1\}^n \to \{0,1\}^n$, it follows that $I^p$, for a uniform random permutation $p : \{0,1\}^n \to \{0,1\}^n$, is indistinguishable from a uniform random function $F : \{0,1\}^n \to \{0,1\}^n$. This is because, no distinguisher that makes a bounded number of queries to an oracle $A : \{0,1\}^n \to \{0,1\}^n$ which is either a uniform random function $f : \{0,1\}^n \to \{0,1\}^n$ or a uniform random permutation $p : \{0,1\}^n \to \{0,1\}^n$, can distinguish between them except with some small probability. This is sometimes called as the "PRF/PRP switching lemma" [CN08b][BR06].

So, since the construction $I^f$, for a uniform random $f : \{0,1\}^n \to \{0,1\}^n$, is indifferentiable from a uniform random function $F : \{0,1\}^n \to \{0,1\}^n$, it is natural to ask if $I^p$, for a uniform random permutation $p : \{0,1\}^n \to \{0,1\}^n$ (together with its inverse denoted by $p^{-1}$), is indifferentiable from a uniform random function $F : \{0,1\}^n \to \{0,1\}^n$.

In other words, can we come up with an efficient simulator $S$, that has oracle access to $F$ and tries to simulate $p, p^{-1}$ (in response to a polynomial number of queries made to it) such that no distinguisher can distinguish between $(I^p, \{p, p^{-1}\})$ and $(F, S^F)$ except with some small probability? Unfortunately, the answer is a negative one, because we can come up with a distinguisher which distinguishes $(I^p, \{p, p^{-1}\})$ and $(F, S^F)$ with overwhelming probability for any efficient simulator $S$.

**Theorem 8.1.** *Let $X_n$ be the uniform distribution on $\mathbb{F}_n^n$ and $Y_n$ be a uniform distribution on $\mathbb{P}_n$, for every positive even integer $n$. Let $\mathcal{X} = \{X_n\}$ and $\mathcal{Y} = \{Y_n\}$. Then, the construction $I$ having oracle access to the distribution family $\mathcal{Y}$ is not indifferentiable from the distribution family $\mathcal{X}$.*

*Proof.* Fix $n$. Here we define a distinguisher $D$ which distinguishes $(I^p, \{p, p^{-1}\})$ and $(F, S^F)$ with overwhelming probability for any efficient simulator $S$, where $p$ is a uniform random permutation mapping $\{0,1\}^n$ to $\{0,1\}^n$ and $p^{-1}$ denotes its inverse and $F$ is a uniform random function mapping $\{0,1\}^n$ to $\{0,1\}^n$. $D$ has access to oracles $X : \{0,1\}^n \to \{0,1\}^n$ and $Y_0, Y_1 : \{0,1\}^n \to \{0,1\}^n$, which are either $(I^p, \{p, p^{-1}\})$ or $(F, S^F)$. $D$ works as follows,

1. Choose arbitrary $a \in \{0,1\}^{n/2}$ and $b$ uniformly at random from $\{0,1\}^{n/2}$.
2. Let $\alpha \leftarrow X(ab)$.
3. Let $\beta \leftarrow Y_1(\alpha)$.
4. Let $\gamma \leftarrow Y_0(a0^{n/2})$.
5. If $\gamma \oplus b0^{n/2} = \beta$ accept, else reject.

It is easy to see that $D$ accepts $(I^p, \{p, p^{-1}\})$ with probability 1. This is because, for any $\{p, p^{-1}\}$, the following holds. $\alpha = I^p(ab)$; $\beta = p^{-1}(\alpha)$, so $\alpha = p(\beta)$. $\gamma = p(a0^{n/2})$; $p(\gamma \oplus b0^{n/2}) = p(p(a0^{n/2}) \oplus b0^{n/2}) = I^p(ab) = \alpha = p(\beta)$; So $\gamma \oplus b0^{n/2} = \beta$. However, $D$ accepts $(F, S^F)$ with probability at most $\frac{q}{2^n}$ for any simulator $S$ making at most $q$ queries to $F$ because, for a random $F$ and random $b$, the equality $\gamma \oplus b0^{n/2} = \beta$ holds only with probability at most $\frac{q}{2^n}$ for any simulator $S$ that makes at most $q$ oracle queries to $F$. This is due to the following reasons. Since $F$ is random and $ab$ is the first query made to $F$, $\alpha$ is random. The simulator sees only $\alpha$ and $a$ and has no idea about $b$. Yet, it has to output $\beta, \gamma$ such that $\gamma \oplus \beta = b0^{n/2}$. So, effectively it has to guess $b$, of which it has no idea about. This can be done with probability at most $\frac{q}{2^n}$, for any simulator $S$ that makes at most $q$ oracle queries to $F$. Hence, we have that, for any simulator $S$ that makes at most $q$ oracle queries to $F$,

$$|Pr\,[D^{I^p, \{p, p^{-1}\}}(1^n) = 1] - Pr\,[D^{F, S^F}(1^n) = 1]| \geq 1 - \frac{q}{2^n}$$

Hence, we conclude that $D$ distinguishes $(I^p, \{p, p^{-1}\})$ and $(F, S^F)$ with overwhelming probability for any efficient simulator $S$. □

## 8.1 Some Ideas on How to Make the Construction Work

### 8.1.1 A Modified Construction

We would like to modify the construction $I^p$ a little bit, so that we can get an indifferentiability theorem out of it. A trick would be to drop a certain amount of bits (say $n/2$ bits) from the output of $I^p$. A reason to believe why this would work is that, this construction is a special case of the sponge construction [BDPVA11a]. Consider the following construction which processes two blocks of input.

**Definition 8.2.** *Let $f$ be a function such that $f : \{0,1\}^n \to \{0,1\}^n$ for every positive even integer $n$. For every positive even integer $n$, define $trI^f : \{0,1\}^n \to \{0,1\}^{n/2}$ as follows*

$$trI^f(xy) = \text{ first } n/2 \text{ bits of } I^f(xy)$$

*where $x, y \in \{0,1\}^{n/2}$.*

Bertoni *et al.* in [BDPVA08] have given an indifferentiablity proof of the sponge construction and since the function $trI^p$ is a special case of the sponge construction, the following theorem is true.

**Theorem 8.3.** *Let $X_n$ be the uniform distribution on $\mathbb{F}_n^{n/2}$ and $Y_n$ be a uniform distribution on $\mathbb{P}_n$, for every positive even integer $n$. Let $\mathcal{X} = \{X_n\}$ and $\mathcal{Y} = \{Y_n\}$. Then, the construction $trI$ having oracle access to the distribution family $\mathcal{Y}$ is indifferentiable from the distribution family $\mathcal{X}$.*

Again, we can generalize this construction so that it processes inputs of arbitrary length and state an indifferentiability theorem about it, similar to the one above. A part of our future work is to give a rigorous proof for this theorem using the technique we used in the previous section.

### 8.1.2 Sequential Indifferentiability

If you consider the distinguisher we gave in previous subsection, it was quite important that the distinguisher is non-sequential, else the attack wouldn't have gone through. So, we think that the following sequential indifferentiability theorem is true about the construction $I^p$ for a random permutation $p$.

**Theorem 8.4.** *Let $X_n$ be the uniform distribution on $\mathbb{F}_n^n$ and $Y_n$ be a uniform distribution on $\mathbb{P}_n$, for every positive even integer $n$. Let $\mathcal{X} = \{X_n\}$ and $\mathcal{Y} = \{Y_n\}$. Then, the construction $I$ having oracle access to the distribution family $\mathcal{Y}$ is sequentially indifferentiable from the distribution family $\mathcal{X}$.*

In our future work, we would like to prove this theorem and also a similar sequential indifferentiability theorem about the more general construction construction processing inputs of arbitrary length, using the technique we used in the previous section.

## 9 Conclusion

In this paper, we used a simple yet rigorous proof technique for proving indifferentiability theorems, by generalizing the indistinguishability proof technique used by Bernstein in [Ber05] to the indifferentiability setting. We used this technique to prove the indifferentiability of the construction $I^f$ (for a uniform random $f : \{0,1\}^n \to \{0,1\}^n$) from a uniform random $F : \{0,1\}^n \to \{0,1\}^n$. Of course, this construction is not very useful in practice since it processes only two blocks of input. We merely used it to test and in the process, elucidate, the proof technique.

As a part of our future work, we would like to extend this proof technique to prove the indifferentiability theorem of a more general version of $I^f$, the one which processes inputs of arbitrary length. We would also like to use our technique to prove the two theorems we stated in the previous section: the one regarding the sequential indifferentiability of $I^p$, for a uniform random permutation $p$ (together with its inverse) and the one regarding the indifferentiability of $trI^p$, for a uniform random permutation $p$ (together with its inverse). Again, these last two constructions are not very useful in practice since they process only two blocks of input. However, proving the indifferentiability of these simple constructions will give us confidence to prove the indifferentiability of the more general versions of these constructions, the ones which process inputs of arbitrary length. Finally, we aim to further simplify the proof of indifferentiability of $I^f$ (for a uniform random $f$) that we presented, and also apply this proof technique to prove the indifferentiability results of various other cryptographic constructions.

**Acknowledgements.** I would like to thank my advisor Dr. Charles Rackoff for suggesting the problem, having numerous invaluable discussions which resulted in this paper, and also for carefully reading the paper and suggesting critical changes. I would also like to thank Dr. Vinod Vaikuntanathan for his helpful feedback.

## References

[AMP12] Elena Andreeva, Bart Mennink, and Bart Preneel, *The parazoa family: generalizing the sponge hash functions.*, Int. J. Inf. Sec. **11** (2012), no. 3, 149–165.

[BDPVA08] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche, *On the Indifferentiability of the Sponge Construction*, Proceedings of the Theory and Applications of Cryptographic Techniques 27th Annual International Conference on Advances in Cryptology (Berlin, Heidelberg), EUROCRYPT'08, Springer-Verlag, 2008, pp. 181–197.

[BDPVA11a] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche, *Cryptographic sponge functions*, `http://sponge.noekeon.org/CSF-0.1.pdf`, 2011.

[BDPVA11b] ———, *The Keccak reference*, `http://keccak.noekeon.org/Keccak-reference-3.0.pdf`, 2011.

[Ber05] D.J. Bernstrein, *A short proof of the unpredictability of cipher block chaining*, `http://cr.yp.to/antiforgery/easycbc-20050109.ps`, 2005.

[BKR00] Mihir Bellare, Joe Kilian, and Phillip Rogaway, *The security of the cipher block chaining message authentication code*, Journal of Computer and System Sciences, Springer-Verlag, 2000, pp. 362–399.

[BM84] Manuel Blum and Silvio Micali, *How to Generate Cryptographically Strong Sequences of Pseudo-random Bits*, SIAM J. Comput. **13** (1984), no. 4, 850–864.

[BPR05] M. Bellare, K. Pietrzak, and P. Rogaway, *Improved security analyses for CBC MACs*, In Advances in Cryptology Crypto 2005, LNCS 3621, Springer-Verlag, 2005, pp. 527–545.

[BR93] Mihir Bellare and Phillip Rogaway, *Random Oracles Are Practical: A Paradigm for Designing Efficient Protocols*, Proceedings of the 1st ACM Conference on Computer and Communications Security (New York, NY, USA), CCS '93, ACM, 1993, pp. 62–73.

[BR95] Mihir Bellare and Phillip Rogaway, *Optimal Asymmetric Encryption How to Encrypt with RSA*, Springer-Verlag, 1995, pp. 92–111.

[BR06] ———, *Code-Based Game-Playing Proofs and the Security of Triple Encryption*, Advances in Cryptology EUROCRYPT 2006, Springer-Verlag, 2006.

[CDMP05] Jean-Sbastien Coron, Yevgeniy Dodis, Cécile Malinaud, and Prashant Puniya, *Merkle Damgård revisited: How to construct a hash function*, Springer-Verlag, 2005, pp. 430–448.

[CGH04] Ran Canetti, Oded Goldreich, and Shai Halevi, *The Random Oracle Methodology, Revisited*, J. ACM **51** (2004), no. 4, 557–594.

[CN08a] Donghoon Chang and Mridul Nandi, *Improved Indifferentiability Security Analysis of chopMD Hash Function.*, FSE, Lecture Notes in Computer Science, vol. 5086, Springer, 2008, pp. 429–443.

[CN08b] ———, *A Short Proof of the PRP/PRF Switching Lemma*, `https://eprint.iacr.org/2008/078`, 2008.

[CPS08] Jean-Sbastien Coron, Jacques Patarin, and Yannick Seurin, *The Random Oracle Model and the Ideal Cipher Model Are Equivalent.*, CRYPTO (David Wagner, ed.), Lecture Notes in Computer Science, vol. 5157, Springer, 2008, pp. 1–20.

[FOPS04] Eiichiro Fujisaki, Tatsuaki Okamoto, David Pointcheval, and Jacques Stern, *RSA-OAEP Is Secure Under the RSA Assumption*, J. Cryptol. **17** (2004), no. 2, 81–104.

[GR10] Sergey Gorbunov and Charles Rackoff, *On the Security of Cipher Block Chaining Message Authentication Code*, `http://people.csail.mit.edu/sergeyg/publications/securityOfCBC.pdf`, 2010.

[HKT10] Thomas Holenstein, Robin Künzler, and Stefano Tessaro, *Equivalence of the Random Oracle Model and the Ideal Cipher Model, Revisited*, CoRR **abs/1011.1264** (2010).

[HR10] Viet Tung Hoang and Phillip Rogaway, *On Generalized Feistel Networks*, Proceedings of the 30th Annual Conference on Advances in Cryptology (Berlin, Heidelberg), CRYPTO'10, Springer-Verlag, 2010, pp. 613–630.

[JK03] J. Jonsson and B. Kaliski, *Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1*, 2 2003.

[Lin92] Torgny Lindvall, *Lectures on the Coupling Method*, Wiley, New York, 1992.

[LR88] Michael Luby and Charles Rackoff, *How to Construct Pseudorandom Permutations from Pseudorandom Functions*, SIAM J. Comput. **17** (1988), no. 2, 373–386.

[Mau02] Ueli Maurer, *Indistinguishability of Random Systems*, In Advances in Cryptology  EUROCRYPT 02, volume 2332 of LNCS, Springer-Verlag, 2002, pp. 110–132.

[Mir02] Ilya Mironov, *(Not so) Random Shuffles of RC4*, Proceedings of the 22Nd Annual International Cryptology Conference on Advances in Cryptology (London, UK, UK), CRYPTO '02, Springer-Verlag, 2002, pp. 304–319.

[MPS12] Avradip Mandal, Jacques Patarin, and Yannick Seurin, *On the Public Indifferentiability and Correlation Intractability of the 6-round Feistel Construction*, Proceedings of the 9th International Conference on Theory of Cryptography (Berlin, Heidelberg), TCC'12, Springer-Verlag, 2012, pp. 285–302.

[MRH03] Ueli Maurer, Renato Renner, and Clemens Holenstein, *Indifferentiability, Impossibility Results on Reductions, and Applications to the Random Oracle Methodology*, Theory of Cryptography - TCC 2004, Lecture Notes in Computer Science, Springer-Verlag, 2003, pp. 21–39.

[MRS09] Ben Morris, Phillip Rogaway, and Till Stegers, *How to Encipher Messages on a Small Domain*, Proceedings of the 29th Annual International Cryptology Conference on Advances in Cryptology (Berlin, Heidelberg), CRYPTO '09, Springer-Verlag, 2009, pp. 286–302.

[MT07] Ueli Maurer and Stefano Tessaro, *Domain extension of public random functions: Beyond the birthday barrier*, In Advances in Cryptology  CRYPTO 07 (2007), Lecture Notes in Computer Science, Springer-Verlag, 2007, pp. 187–204.

[PR00] Erez Petrank and Charles Rackoff, *CBC MAC for Real-Time Data Sources.*, J. Cryptology **13** (2000), no. 3, 315–338.

[Sho01] Victor Shoup, *OAEP Reconsidered*, Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology (London, UK, UK), CRYPTO '01, Springer-Verlag, 2001, pp. 239–259.

[Tho00] Hermann Thorisson, *Coupling, Stationarity, and Regeneration (Probability and Its Applications)*, Springer, New York, 2000.

[Vau03] Serge Vaudenay, *Decorrelation: a theory for block cipher security*, Journal of Cryptology **16** (2003), 2003.

## A A Flaw in the Distinguishing Attack of [CPS08] for the 5-Round Feistel Network

We show an efficient simulator that passes the distinguishing attack of [CPS08] for the 5-round Feistel network. We first give a sketch of their attack and then we give an efficient simulator that passes the test. We also give a simple fix to the attack so that their claim becomes true.

**Distinguishing attack of [CPS08]** Fix $n$. The distinguisher is provided 7 oracles $p_0$,$p_1$ : $\{0,1\}^{2n} \to \{0,1\}^{2n}$ and $h_1$,$h_2$,$h_3$,$h_4$,$h_5$ : $\{0,1\}^n \to \{0,1\}^n$ (where the oracles are either $(\Phi_{5,F_5}, \Phi^{-1}_{5,F_5}), (F_5)$ or $(P, P^{-1}), (S^{P,P^{-1}})$). The attack proceeds as follows.

1. Choose $y, y', z \in \{0,1\}^n$ arbitrarily.
2. Query $h_3(y), h_3(y')$.
3. Let $x = h_3(y) \oplus z$, $x' = h_3(y') \oplus z$.
4. Query $h_2(x)$, $h_2(x')$.
5. Let $R_0 = y \oplus h_2(x)$, $R_1 = y \oplus h_2(x')$, $R_2 = y' \oplus h_2(x')$, $R_3 = y' \oplus h_2(x)$.
6. Query $h_1(R_0)$, $h_1(R_1)$,$h_1(R_2)$, $h_1(R_3)$.
7. Let $L_0 = h_1(R_0) \oplus x$, $L_1 = h_1(R_1) \oplus x'$, $L_2 = h_1(R_2) \oplus x'$, $L_3 = h_1(R_3) \oplus x$.
8. Let $S_0T_0 \leftarrow p_0(L_0R_0)$, $S_1T_1 \leftarrow p_0(L_1R_1)$, $S_2T_2 \leftarrow p_0(L_2R_2)$, $S_3T_3 \leftarrow p_0(L_3R_3)$
9. If $R_0 \oplus R_1 \oplus R_2 \oplus R_3 = 0$ and $S_0 \oplus S_1 \oplus S_2 \oplus S_3 = 0$ accept, else reject.

Coron *et al.* argued that $D$ accepts $((\Phi_{5,F_5}, \Phi^{-1}_{5,F_5}), F_5)$ with probability 1 for any 5-tuple of functions $F_5$. However, for a random $P, P^{-1}$, $D$ accepts $(P, P^{-1}), (S^{P,P^{-1}})$ with negligible probability for any efficient simulator $S$. However, we can come up with an efficient simulator $S$ such that for a random $P, P^{-1}$, $D$ accepts $(P, P^{-1}), (S^{P,P^{-1}})$ with probability 1. The simulator $S$ outputs a 5-tuple of constant functions (each of which say, always output 0 on any input). So when the 7 oracles ($p_0$,$p_1$),($h_1$,$h_2$,$h_3$,$h_4$,$h_5$) are $(P, P^{-1}), (S^{P,P^{-1}})$, for every $i$, $1 \le i \le 5$, the oracle $h_i$ always outputs 0 on any input. It is clear that $S$ is efficient. We prove that for a random $P, P^{-1}$, $D$ accepts $(P, P^{-1}), (S^{P,P^{-1}})$ with probability 1. Analysing the above attack for this case,

1. We have that $x = h_3(y) \oplus z = z$, $x' = h_3(y') \oplus z = z$ and hence $x = x' = z$.
2. We have that $R_0 = y \oplus h_2(x) = y$, $R_1 = y \oplus h_2(x') = y$, $R_2 = y' \oplus h_2(x') = y'$, $R_3 = y' \oplus h_2(x) = y'$ and hence $R_0 = R_1 = y$ and $R_2 = R_3 = y'$.
3. We have that $L_0 = h_1(R_0) \oplus x = x$, $L_1 = h_1(R_1) \oplus x' = x'$, $L_2 = h_1(R_2) \oplus x' = x'$, $L_3 = h_1(R_3) \oplus x = x$. Hence we have that $L_0 = L_1 = L_2 = L_3 = z$ (since $x = x' = z$).
4. We have that $S_0T_0 = p_0(L_0R_0) = p_0(zy)$, $S_1T_1 = p_0(L_1R_1) = p_0(zy)$, $S_2T_2 = p_0(L_2R_2) = p_0(zy')$, $S_3T_3 = p_0(L_3R_3) = p_0(zy')$.
5. Hence we have that $S_0T_0 = S_1T_1$ and $S_2T_2 = S_3T_3$
6. Hence it is clear that $R_0 \oplus R_1 \oplus R_2 \oplus R_3 = 0$ and $S_0 \oplus S_1 \oplus S_2 \oplus S_3 = 0$ (from points 2 and 5).

So it is clear that for any $P, P^{-1}$, $D$ accepts $(P, P^{-1}), (S^{P,P^{-1}})$ with probability 1 and hence the attack doesn't go through. The simplest way to fix the attack is to make sure that the $R_i$s are distinct. That is, in step 5 of the attack, we should include the check that if for some $i, j$, $1 \le i \ne j \le 4$, $R_i = R_j$, then reject. It is easy to argue that this modified distinguishing attack goes through and we skip the details.