On the Public Indifferentiability and Correlation Intractability of the 6-Round Feistel Construction

Avradip Mandal[1], Jacques Patarin[2], and Yannick Seurin[3]

[1] University of Luxembourg
[2] University of Versailles, France
[3] ANSSI, Paris, France
avradip.mandal@uni.lu,jacques.patarin@uvsq.fr,yannick.seurin@m4x.org

Abstract. We show that the Feistel construction with six rounds and random round functions is *publicly* indifferentiable from a random invertible permutation (a result that is not known to hold for full indifferentiability). Public indifferentiability (*pub-indifferentiability* for short) is a variant of indifferentiability introduced by Yoneyama *et al.* [YMO09] and Dodis *et al.* [DRS09] where the simulator knows all queries made by the distinguisher to the primitive it tries to simulate, and is useful to argue the security of cryptosystems where all the queries to the ideal primitive are public (as *e.g.* in many digital signature schemes). To prove the result, we introduce a new and simpler variant of indifferentiability, that we call sequential indifferentiability (*seq-indifferentiability* for short) and show that this notion is in fact equivalent to pub-indifferentiability for stateless ideal primitives. We then prove that the 6-round Feistel construction is seq-indifferentiable from a random invertible permutation. We also observe that sequential indifferentiability implies correlation intractability, so that the Feistel construction with six rounds and random round functions yields a correlation intractable invertible permutation, a notion we define analogously to correlation intractable functions introduced by Canetti *et al.* [CGH98].

Keywords: indifferentiability, correlation intractability, Feistel construction

1 Introduction

Indifferentiability. Indifferentiability has been introduced by Maurer *et al.* [MRH04] as a generalization of the concept of indistinguishability for systems using *public* components (*i.e.* components that can be queried by any party including the adversary). This framework has since then gained much popularity, and starting with [CDMP05] it has been widely used to analyze hash functions built from a smaller ideal primitive, *e.g.* a fixed input-length (FIL) random compression function or an ideal block cipher [BR06,BDPA08,CLNY06,CN08,DRRS09,FLP08,HPY07,MT07]. Informally, a construction $\mathcal{C}$ using an ideal primitive $\boldsymbol{F}$ (*e.g.* a hash function based on a FIL random compression function) is said to be indifferentiable from another ideal primitive $\boldsymbol{G}$ (*e.g.* a random oracle) if there exists a simulator $\mathcal{S}$ accessing $\boldsymbol{G}$ such that the two systems $(\boldsymbol{G}, \mathcal{S}^{\boldsymbol{G}})$ and $(\mathcal{C}^{\boldsymbol{F}}, \boldsymbol{F})$ are indistinguishable. Roughly, the goal of the simulator is twofold: it must provide answers that are consistent with $\boldsymbol{G}$, without deviating too much from the distribution of answers of $\boldsymbol{F}$. Indifferentiability allows modular proofs of security in idealized models in the sense that if a construction $\mathcal{C}^{\boldsymbol{F}}$ is indifferentiable from an ideal primitive $\boldsymbol{G}$, then any cryptosystem proven secure when used with $\boldsymbol{G}$ remains secure when used with the construction $\mathcal{C}^{\boldsymbol{F}}$.[4] For example, if a cryptosystem is secure in the random oracle model, and some hash function construction $H^{\boldsymbol{f}}$ based on a FIL random compression function $\boldsymbol{f}$ is indifferentiable from a random oracle, then the cryptosystem is still secure when used with $H^{\boldsymbol{f}}$. More interestingly from a theoretical point of view, Coron *et al.* [CDMP05] showed that a number of variants of the Merkle-Damgård construction [Dam89,Mer89], used with an ideal cipher in Davies-Meyer

[4] It was recently pointed out that this composition theorem only holds for cryptosystems whose security is defined by so called *single-stage games* [RSS11].

mode [PGV93,BRS02], are indifferentiable from a random oracle. This implies that any functionality that can be securely implemented in the random oracle model can also be securely realized in the ideal cipher model.

The Feistel construction with public round functions. The Feistel construction turns a function F from n-bit strings to n-bit strings into an (efficiently invertible) permutation on $2n$-bit strings. It is computed as $\Psi^F(L,R) = (R, L \oplus F(R))$. In their seminal paper [LR88] which triggered a lot of subsequent work [Mau92,NR99,Pat90,Pat91,Pat98,Pat03,Pat04,Vau03], Luby and Rackoff showed that three (resp. four) rounds of the Feistel construction, with independent pseudorandom functions in each round, yields a pseudorandom permutation (resp. strong pseudorandom permutation). The core of this result is in fact purely information-theoretic [Mau92], meaning that the Feistel construction with three (resp. four) rounds and random round functions is indistinguishable from a random permutation (resp. an invertible random permutation) by any *computationally unbounded* distinguisher limited to a *polynomial number of oracle queries*. The Luby-Rackoff theorem crucially relies on the secrecy of the round functions. A few papers studied what happens when the round functions are made public. In particular, Ramzan and Reyzin [RR00] have shown that the Feistel construction with four rounds remains strongly pseudorandom even when the distinguisher has oracle access to the two middle round functions (but not to the first or the fourth round function). Dodis and Puniya [DP07] have studied various properties of the Feistel construction (unpredictability, pseudorandomness) when all intermediate round values of the Feistel computation are leaked to the adversary and shown that in that case a super-logarithmic number of rounds was necessary and sufficient for the property to be inherited by the Feistel construction from the round functions.

Indifferentiability of the Feistel construction. As already mentioned, it is possible to securely instantiate a random oracle in the ideal cipher model. A natural question is whether the other direction holds, namely whether there is a construction using a random oracle that securely implements a random invertible permutation.[5] Given its numerous cryptographic properties, the Feistel construction (with public random round functions) appears as an obvious candidate for this task. Again, this question can be rigorously formulated in the indifferentiability framework: namely, is the Feistel construction with sufficiently many rounds, and public random round functions, indifferentiable from a random invertible permutation? Dodis and Puniya [DP06] considered the problem in the so-called *honest-but-curious* model, where the distinguisher only sees the queries made by the Feistel construction to the random round functions, but is not allowed to make arbitrary queries to the round functions. In this setting, they showed that a super-logarithmic number of rounds is sufficient to securely realize a random invertible permutation. However, since full indifferentiability is not implied in general by indifferentiability in the honest-but-curious model (these two notions are in fact incomparable [CPS08]), they were not able to conclude in the general setting. Coron, Patarin, and Seurin [CPS08] gave a first proof that the Feistel construction with six rounds is indifferentiable from a random invertible permutation. The proof was rather involved, and Künzler [Kün09] later found a distinguishing attack against the simulator given in [CPS08], therefore invalidating the indifferentiability proof.[6] Only recently, Holenstein *et al.* [HKT11] gave a new proof that the Feistel construction with *fourteen* rounds is indifferentiable from a random invertible permutation, which was inspired from a previous proof for ten rounds that appeared in the PhD thesis of Seurin [Seu09] but had some gaps.

Public indifferentiability. Yoneyama *et al.* [YMO09] and Dodis *et al.* [DRS09] independently realized that indifferentiability was sometimes stronger than needed to argue security of cryptosystems. In particular, when all queries made to the ideal primitive are public (like in many digital signature schemes such as FDH [BR93], probabilistic FDH [Cor02], PSS [BR96]..., where all queries to the hash function can be revealed to the attacker without affecting the security), the weaker notion of *public* indifferentiability is sufficient. [YMO09,DRS09] were both concerned with indifferentiability from a

[5] Such a construction easily implies a secure ideal cipher by simply prepending the key of the block cipher to the input of each random oracle queries.

[6] We stress that this does not mean that the 6-round Feistel construction is not indifferentiable from a random invertible permutation, but only that no one is able to give a proof at the moment.

random oracle and respectively called this notion *leaky random oracle* and *public-use random oracle*. Public indifferentiability is defined similarly to indifferentiability, but the task of the simulator is made easier by letting it know all queries made by the distinguisher to the ideal primitive $\boldsymbol{G}$.

Correlation intractability. Correlation intractability was introduced by Canetti *et al.* [CGH98] as an attempt to capture as many security properties of the random oracle as possible. A family of functions is said to be correlation intractable if for a random function of the family it is hard to find a sequence of inputs that together with their image satisfy a relation that would be hard to satisfy for a uniformly random function (a so-called *evasive* relation). Correlation intractability in particular implies collision resistance, pre-image resistance and many other security properties usually required for cryptographic hash functions. Unfortunately, Canetti *et al.* also showed that in the standard model, no correlation intractable hash function family exists. A consequence of this non-existence result is that there are cryptosystems that are secure in the random oracle model, but insecure when the random oracle is instantiated by any function family. Though correlation intractability was primarily defined in the standard model, it is easily transposable to idealized models. As we will see our result establishes a connection between correlation intractability and public indifferentiability.

Contributions of this work. We define a new and weaker notion of indifferentiability that we call *sequential* indifferentiability (*seq-indifferentiability* for short). This new definition only restricts *the order* in which the distinguisher can query the two oracles it is granted access to: it can first query the primitive $\boldsymbol{F}$ (or the simulator $\mathcal{S}$), and then the construction $\mathcal{C}^{\boldsymbol{F}}$ (or the ideal primitive $\boldsymbol{G}$), but not $\boldsymbol{F}/\mathcal{S}$ again. We show that when the ideal primitive $\boldsymbol{G}$ is stateless (which is the most usual case), this notion is equivalent to *public* indifferentiability introduced by [DRS09,YMO09] where all queries to the primitive $\boldsymbol{G}$ are public. However the seq-indifferentiability notion has the advantage of being simpler and easier to use in proofs. This simple restriction on the queries of the distinguisher enables to give a relatively simple proof that the 6-round Feistel construction with random round functions is seq-indifferentiable (and hence also publicly indifferentiable) from a random invertible permutation, a result whose analogue for full indifferentiability seems out of reach at the moment. Our result in particular implies that any scheme proven secure in the random invertible permutation model or the ideal cipher model and where all queries to the ideal primitive can be made public without affecting the security (*e.g.* signature schemes like OPSSR [Gra02] and subsequent variants [KW03,CMPP05]) remains secure in the random oracle model when using a 6-round Feistel construction (while the best generic replacement previously to our work was the 14-round Feistel construction [HKT11]).

Though weaker than full indifferentiability, we also show that seq-indifferentiability is still sufficiently strong to imply correlation intractability. In particular, our result shows that the 6-round Feistel construction with random round functions yields a correlation intractable invertible permutation (we note that previous observations [CPS08] already implied that the 5-round Feistel construction fails to provide a correlation intractable invertible permutation). We discuss the implications of this result for chosen-key and known-key attacks on block ciphers [KR07].

On a slightly different topic, we also analyze the Feistel-like domain extension construction for ideal ciphers proposed by Coron *et al.* [CDMS10] and show that in the seq-indifferentiability model one can obtain a security bound beyond the birthday barrier.

Open problems. The most challenging open question is of course whether the 6-round Feistel construction is fully indifferentiable from a random invertible permutation, and if not, what is the minimal number of rounds needed to achieve this property. We hope that our result will constitute a first step towards a finer understanding of this question. In particular, our result implies that if the 6-round Feistel construction is *not* fully indifferentiable from a random invertible permutation, then this cannot be shown by proving that it is not correlation intractable as was done for five rounds. Another interesting problem is to weaken the assumptions on the round functions and see which property would continue to hold: *e.g.* is the 6-round Feistel construction with correlation intractable round functions still a correlation intractable invertible permutation? A related question is whether our result could be a first step towards proposing plausible constructions of (restricted) correlation intractable function families in the standard model, a question left open by [CGH98, Section 5.1].

Organization. In Section 2, we start by giving the definition of sequential indifferentiability and prove that it is equivalent to public indifferentiability for stateless ideal primitives. In Section 3, we prove the main result of this paper, namely that the 6-round Feistel construction is sequentially (and hence publicly) indifferentiable from a random invertible permutation. In Section 4, we apply this result to prove the correlation intractability of the 6-round Feistel construction.

2 Preliminaries

2.1 Notations and Definitions

Notations. $[i..j]$ will denote the set of integers k such that $i \leq k \leq j$. We will use n to denote the security parameter, and in sections dealing with the Feistel construction we will identify n with the input and output length of the round functions. We will write $f \in \mathtt{poly}(n)$ to denote a polynomially bounded function and $f \in \mathtt{negl}(n)$ to denote a negligible function. When $\mathcal{X}$ is a non-empty finite set, we write $x \leftarrow_{\mathcal{R}} \mathcal{X}$ to mean that a value is sampled uniformly at random from $\mathcal{X}$ and assigned to x. PPT will stand for probabilistic polynomial-time, and ITM for interactive Turing machine.

Ideal primitives. Given two sets $\mathtt{Dom} \subset \{0,1\}^*$ and $\mathtt{Rng} \subset \{0,1\}^*$, we denote $\mathcal{F}(\mathtt{Dom}, \mathtt{Rng})$ the set of all functions from $\mathtt{Dom}$ to $\mathtt{Rng}$. A primitive $\mathbb{G}$ is a sequence $\mathbb{G} = (\mathtt{Dom}_n, \mathtt{Rng}_n, \mathbb{G}_n)_{n\in\mathbb{N}}$ where $\mathbb{G}_n \subset \mathcal{F}(\mathtt{Dom}_n, \mathtt{Rng}_n)$. The ideal primitive $\boldsymbol{G}$ associated with $\mathbb{G}$ is the sequence of random variables $(\boldsymbol{G}_n)_{n\in\mathbb{N}}$ where $\boldsymbol{G}_n$ is uniformly distributed over $\mathbb{G}_n$. We will often adopt the lazy sampling view [BR06] to describe ideal primitives queried as oracles.

A random function $\boldsymbol{F} = (\boldsymbol{F}_n)_{n\in\mathbb{N}}$ is the ideal primitive associated to the set of all functions from $\{0,1\}^n$ to $\{0,1\}^n$. Queried as an oracle it returns a uniformly random string in $\{0,1\}^n$ if x was never queried, or the same answer as before if x was previously queried.

A random invertible permutation $\boldsymbol{P} = (\boldsymbol{P}_n)_{n\in\mathbb{N}}$ is the ideal primitive associated with the sequence $\mathbb{P} = (\mathtt{Dom}_n, \mathtt{Rng}_n, \mathbb{P}_n)_{n\in\mathbb{N}}$ where $\mathtt{Dom}_n = \{0,1\} \times \{0,1\}^n$, $\mathtt{Rng}_n = \{0,1\}^n$, and $\mathbb{P}_n$ is the set of functions P such that $x \mapsto P(0,x)$ is a permutation of $\{0,1\}^n$, and $y \mapsto P(1,y)$ its inverse. Queries of the form $(0,x)$ and $(1,y)$ will be called respectively *forward* and *backward* queries. In the lazy sampling point of view, $\boldsymbol{P}_n$ keeps two lists L_x and L_y of forward and backward queries whose image is already defined together with an invertible mapping from L_x to L_y. Upon receiving a forward query $(0,x)$ such that $x \notin L_x$ it returns an answer y uniformly random over $\{0,1\}^n \setminus L_y$, and adds x to L_x and y to L_y and updates the mapping (and reciprocally for a backward query $(1,y)$). Later, we will occasionally refer to L_x and L_y as the *history* of the random invertible permutation. An ideal cipher $\boldsymbol{E} = (\boldsymbol{E}_n)$ takes an additional input, the key, of length $\ell(n)$, and for each key $k \in \{0,1\}^{\ell(n)}$, $\boldsymbol{E}_n(k,\cdot)$ is an independent random invertible permutation over $\{0,1\}^n$.

A two-sided random function on $\{0,1\}^n$, denoted $\boldsymbol{R}_n$, is very similar to a random invertible permutation. It also keeps to lists L_x and L_y together with an invertible mapping from L_x to L_y. However when receiving a forward query $(0,x)$ such that $x \notin L_x$ or a backward query $(1,y)$ such that $y \notin L_y$, it returns a *uniformly random* answer in $\{0,1\}^n$. In case a collision happens, the previous image or pre-image is removed from L_y or L_x and the mapping is updated accordingly. Note that a two-sided random function is stateful: it may return different answers to the same query (however at any time it defines an invertible mapping from L_x to L_y). A two-sided random function is statistically indistinguishable from a random invertible permutation: the so called PRF/PRP switching lemma [BR06] establishes[7] that an oracle machine making at most q oracle queries can distinguish $\boldsymbol{P}_n$ from $\boldsymbol{R}_n$ with advantage at most $q^2/2^{n+1}$.

In the following, we omit the subscripts when the domain and the range of an ideal primitive are clear from the context. A *construction* will simply be a Turing machine having oracle access to an ideal primitive and implementing another given primitive. The main construction we will consider in this work is the Feistel construction.

[7] Strictly speaking, the result is proven in [BR06] for one-sided functions and permutations, but the proof can be straightforwardly adapted to two-sided primitives.

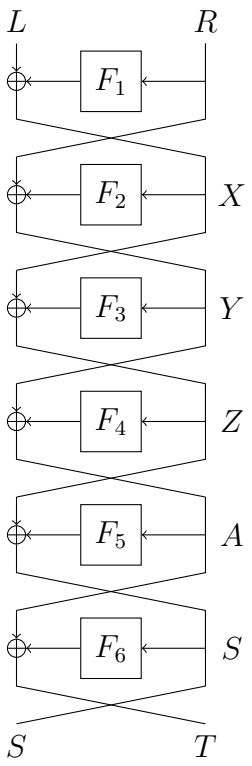

Fig. 1. Notations used for the 6-round Feistel construction.

The Feistel construction. Given a function $F : \{0,1\}^n \to \{0,1\}^n$, the basic (1-round) Feistel construction is the permutation on $\{0,1\}^{2n}$ defined by $\Psi^F(L,R) = (R, L \oplus F(R))$. Its inverse is computed by $(\Psi^F)^{-1}(S,T) = (T \oplus F(S), S)$. (Here L, R, S, and T are n-bit strings). The k-round Feistel construction associated to round functions $(F_1, \ldots, F_k)$ takes inputs $x \in \{0,1\} \times \{0,1\}^{2n}$ and is defined by:

$$\Psi_k^{(F_1,\ldots,F_k)}(0,(L,R)) = \Psi^{F_k} \circ \cdots \circ \Psi^{F_1}(L,R)$$
$$\Psi_k^{(F_1,\ldots,F_k)}(1,(S,T)) = \left(\Psi^{F_1}\right)^{-1} \circ \cdots \circ \left(\Psi^{F_k}\right)^{-1}(S,T) \ .$$

Notations used for denoting the intermediate round values for the 6-round Feistel construction are given in Figure 1. In the following, when considering the Feistel construction using k independent random functions, we will simply note $\boldsymbol{F} = (\boldsymbol{F}_1, \ldots, \boldsymbol{F}_k)$ this tuple of functions and $\Psi_k^{\boldsymbol{F}} = \Psi_k^{(\boldsymbol{F}_1,\ldots,\boldsymbol{F}_k)}$.

2.2 Sequential Indifferentiability

Indifferentiability was originally formulated within the formalism of *random systems* [Mau02]. We adopt here the simpler formulation using interactive Turing machines as in [CDMP05]. We first recall the classical definition of indifferentiability [MRH04]. For this, we slightly change the way one usually measure the cost of queries of a distinguisher (this will make our results simpler to express). Given a distinguisher $\mathcal{D}$, the *total oracle queries cost* of $\mathcal{D}$ is the number of queries received by the oracle $\boldsymbol{F}$ when $\mathcal{D}$ interacts with $(\mathcal{C}^{\boldsymbol{F}}, \boldsymbol{F})$. Hence this is the sum of the number of direct queries of $\mathcal{D}$ to $\boldsymbol{F}$ and the number of queries made by $\mathcal{C}$ to $\boldsymbol{F}$ to answer $\mathcal{D}$'s queries.

Definition 1 ((Statistical, Strong) Indifferentiability). *Let $q, \sigma : \mathbb{N} \to \mathbb{N}$ and $\epsilon : \mathbb{N} \to \mathbb{R}$ be three functions of the security parameter n. A construction $\mathcal{C}$ with oracle access to an ideal primitive $\boldsymbol{F}$ is said to be statistically and strongly (q, σ, ϵ)-indifferentiable from an ideal primitive $\boldsymbol{G}$ if there exists an oracle ITM $\mathcal{S}$ such that for any distinguisher $\mathcal{D}$ of total oracle queries cost at most q, $\mathcal{S}$ makes at most σ oracle queries, and the following holds:*

$$\left|\Pr\left[\mathcal{D}^{\boldsymbol{G},\mathcal{S}^{\boldsymbol{G}}}(1^n) = 1\right] - \Pr\left[\mathcal{D}^{\mathcal{C}^{\boldsymbol{F}},\boldsymbol{F}}(1^n) = 1\right]\right| \leq \epsilon \ .$$

$\mathcal{C}^{\boldsymbol{F}}$ is simply said to be statistically and strongly indifferentiable from $\boldsymbol{G}$ if for any $q \in \mathtt{poly}(n)$, the above definition is fulfilled with $\sigma \in \mathtt{poly}(n)$ and $\epsilon \in \mathtt{negl}(n)$.

Definition 1 does not refer to the running time of $\mathcal{S}$ and $\mathcal{D}$. When only polynomial-time algorithms are considered, indifferentiability is said to be *computational*. Weak indifferentiability is defined as above, but the order of quantifiers for the distinguisher and the simulator are switched (for all distinguisher, there is a simulator...). We will mainly be concerned with statistical strong indifferentiability in this work, but we note that weak indifferentiability is sufficient for our results on correlation intractability in Section 4.

In order to define our new notion of indifferentiability, we will consider a restricted class of distinguisher, called *sequential distinguisher*, which can only make queries in a specific order. Such a distinguisher first queries the primitive $\boldsymbol{F}$ (or the simulator $\mathcal{S}$) as it wishes, and then the construction $\mathcal{C}^{\boldsymbol{F}}$ (or the primitive $\boldsymbol{G}$) as it wishes, but after its first query to $\mathcal{C}^{\boldsymbol{F}}$ or $\boldsymbol{G}$, it cannot query $\mathcal{S}$ or $\boldsymbol{F}$ again. Sequential indifferentiability (*seq-indifferentiability* for short) is defined relatively to such distinguishers (see also Figure 2).

Definition 2 (Seq-indifferentiability). *A construction $\mathcal{C}$ with oracle access to an ideal primitive $\boldsymbol{F}$ is said to be (statistically and strongly) (q, σ, ϵ)-seq-indifferentiable from an ideal primitive $\boldsymbol{G}$ if Definition 1 is fulfilled when $\mathcal{D}$ ranges over the class of sequential distinguishers.*

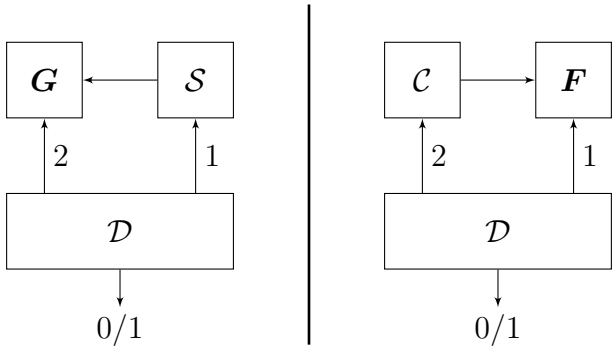

Fig. 2. The sequential indifferentiability notion. The numbers next to query arrows indicate in which order the distinguisher accesses both oracles. After its first query to the left oracle, the distinguisher cannot query the right oracle any more.

Full indifferentiability obviously implies seq-indifferentiability. Yoneyama *et al.* [YMO09] and Dodis *et al.* [DRS09] have introduced another weakened notion of indifferentiability, where the primitive $\boldsymbol{G}$ is only queried on *public* inputs, that we call here public indifferentiability (*pub-indifferentiability* for short). This can be formalized as follows: given an ideal primitive $\boldsymbol{G}$, we define the augmented ideal primitive $\overline{\boldsymbol{G}}$ as the primitive exposing two interfaces: the first (regular) one is the same as $\boldsymbol{G}$, and the

second is an interface `Reveal` that, when queried, returns the ordered sequence of all (regular) queries and corresponding answers made so far by any party to the regular interface. The second interface can only be used by the simulator, not by the distinguisher.

Definition 3 (Pub-indifferentiability). *A construction $\mathcal{C}$ with oracle access to an ideal primitive $\boldsymbol{F}$ is said to be (statistically and strongly) (q, σ, ϵ)-pub-indifferentiable from an ideal primitive $\boldsymbol{G}$ if there exists an oracle ITM $\mathcal{S}$ such that for any distinguisher $\mathcal{D}$ of total oracle queries cost at most q, $\mathcal{S}$ makes at most σ oracle queries, and the following holds:*

$$\left|\Pr\left[\mathcal{D}^{\boldsymbol{G},\mathcal{S}^{\overline{\boldsymbol{G}}}}(1^n)=1\right]-\Pr\left[\mathcal{D}^{\mathcal{C}^{\boldsymbol{F}},\boldsymbol{F}}(1^n)=1\right]\right|\leq\epsilon\ .$$

As explained in [DRS09], the composition theorem of [MRH04] still holds with pub-indifferentiability for cryptosystems where all messages queried to $\boldsymbol{G}$ can be inferred from the adversary's query during the security experiment.

Clearly, pub-indifferentiability implies seq-indifferentiability. Indeed, since after its first query to $\boldsymbol{G}$ a sequential distinguisher never queries the simulator again, the interface `Reveal` is of no use to the simulator. A less trivial result is that seq-indifferentiability implies pub-indifferentiability *for stateless*[8] *ideal primitives* $\boldsymbol{G}$, thus making seq- and pub-indifferentiability equivalent notions in that case.

Theorem 1. *Let $\mathcal{C}$ be a construction with oracle access to some ideal primitive $\boldsymbol{F}$. If $\mathcal{C}^{\boldsymbol{F}}$ is statistically (resp. computationally) strongly $(2q, \sigma, \epsilon)$-seq-indifferentiable from a* stateless *ideal primitive $\boldsymbol{G}$, then $\mathcal{C}^{\boldsymbol{F}}$ is statistically (resp. computationally) strongly $(q, \sigma + q, \epsilon)$-pub-indifferentiable from $\boldsymbol{G}$.*

Proof. The proof is deferred to Appendix A for reasons of space. □

Ristenpart[9] observed that the above theorem does not hold (at least in the computational setting) when $\boldsymbol{G}$ is stateful. This is explained in Appendix B. A very simple example enables to separate full indifferentiability from seq/pub-indifferentiability, namely the Merkle-Damgård construction without strengthening using a random compression function: it was proven in [CDMP05] that it is not indifferentiable from a random oracle (a consequence of length-extension attacks), and in [DRS09] that it is pub-indifferentiable from a random oracle.

2.3 Seq-Indifferentiability Beyond the Birthday Barrier for the Construction of [CDMS10]

In [CDMS10], Coron *et al.* considered the problem of ideal cipher domain extension. They showed that a 3-round Feistel-like construction based on an n-bit ideal cipher is indifferentiable from a $2n$-bit ideal cipher. They obtained a birthday security bound, namely the construction is secure as long as the attacker makes $q \ll 2^{n/2}$ many queries. However, in the same paper it was shown that the construction is actually secure up to $q \ll 2^n$ many queries in the standard indistinguishability model, where the attacker cannot make queries to the smaller n-bit ideal cipher. It was left as an open problem whether obtaining a similar improved security bound in the indifferentiability model was possible. Here we give a partial positive answer to that question, namely showing that the 3-round Feistel-like construction is seq-indifferentiable and pub-indifferentiable from an ideal cipher up to $q \ll 2^n$ queries. Details can be found in Appendix F.

[8] By stateless we mean that the answer of $\boldsymbol{G}$ to any query only depends on the query and the randomness of $\boldsymbol{G}$ and not on any additional state information. In particular, for fixed randomness, $\boldsymbol{G}$ always returns the same answer to a given query.

[9] Personal communication

3 Seq-Indifferentiability of the 6-Round Feistel Construction

In this section we prove the main result of this paper which states that the Feistel construction with 6 rounds and random round functions is seq-indifferentiable from a random invertible permutation, and hence also pub-indifferentiable since a random invertible permutation is stateless. Before stating the result, we recall that in [CPS08], it was shown that the Feistel construction with five rounds is not indifferentiable from a random invertible permutation. In fact, the distinguisher they described is sequential, which implies that the 5-round Feistel construction is not even seq-indifferentiable from a random invertible permutation. We recall this attack in Appendix C.

Theorem 2. *The Feistel construction with six rounds and random round functions is statistically and strongly (q, σ, ϵ)-seq-indifferentiable from a random invertible permutation, where:*

$$\sigma(q) = q^2 \quad \textit{and} \quad \epsilon(q) = \frac{8q^4}{2^n} + \frac{q^4}{2^{2n}} \ .$$

The rest of this section is devoted to the proof of Theorem 2. We will consider a sequential distinguisher $\mathcal{D}$ that first issues at most q_f queries to the simulator (or the random functions $\boldsymbol{F}_i$). These queries will be called F-queries. Then, it issues at most q_p queries to the random permutation $\boldsymbol{P}$ (or the Feistel construction $\Psi_6^{\boldsymbol{F}}$). These queries will be called P-queries. The total oracle queries cost is $q_f + 6q_p$ (for each P-query, the Feistel construction makes 6 F-queries to compute the answer) and is assumed to be less than q.

We start by describing how the simulator $\mathcal{S}$ works. It maintains an history of values for which each round function has been defined (either because this value has been queried by the distinguisher, or because the simulator has set this value internally). We will note F_i, $i \in [1..6]$ the history of the i-th round function, that is a set of pairs $(U, V) \in \{0,1\}^n \times \{0,1\}^n$, where U is an input to round function F_i and V is the corresponding image (which we denote $F_i(U) = V$). We write $U \in F_i$ to denote that the image of U by F_i is defined in the history. Initially round function values $F_i(U)$ are undefined for all $i \in [1..6]$ and all $U \in \{0,1\}^n$. The images are then modified during the execution of the simulator. $F_i(U) \leftarrow V$ means that the image of U by F_i is set to V and $F_i(U) \leftarrow_{\mathcal{R}} \{0,1\}^n$ means that the image of U by F_i is set uniformly at random in $\{0,1\}^n$. If a round function value is already in the history and a new assignment occurs, the previous value is overwritten (alternatively, we could let the simulator abort in this case, as in [CPS08], but as we will see this happens only with negligible probability so that the exact behavior of the simulator in such a case in unessential). We will note $\mathcal{H} = (F_1, \ldots, F_6)$ the complete history of the six round functions.

When the simulator receives a F-query (i, U) (meaning that the distinguisher asks for the image of U through round function F_i), it calls an internal procedure `Query`(i, U). This procedure checks whether the corresponding image is in the history of F_i, in which case it returns this value and stops. Otherwise it sets the image uniformly at random. If $i = 1$, 2, 5, or 6, it does nothing more. If $i = 3$ or 4, the simulator additionally completes all centers $(Y, Z) \in F_3 \times F_4$ newly created so that the corresponding values of (L, R) and (S, T) obtained by evaluating the Feistel construction respectively backward and forward are consistent with the random permutation $\boldsymbol{P}$, meaning that $\boldsymbol{P}(0, (L, R)) = (S, T)$. This is done by calling two internal procedures `CompleteForward` (if $i = 4$) or `CompleteBackward` (if $i = 3$) which "adapts" two round function values ($F_5(A)$ and $F_6(S)$ for `CompleteForward`, and $F_1(R)$ and $F_2(X)$ for `CompleteBackward`) so that the Feistel matches with the random permutation. The pseudo-code for the three procedures is given below. Statements put in boxes in `CompleteForward` and `CompleteBackward` are replacements for a different system used in the indifferentiability proof and can be ignored for the moment.

There are two points to prove in order to obtain Theorem 2: that the simulator runs in polynomial time, and then that the probabilities that the distinguisher outputs 1 when interacting with $(\boldsymbol{P}, \mathcal{S}^{\boldsymbol{P}})$ and $(\Psi_6^{\boldsymbol{F}}, \boldsymbol{F})$ differ by a negligible quantity ϵ. The following lemma shows that the simulator runs in time polynomial in the number of queries it receives.

Algorithm 1 Simulator

variable: round function histories $F_1, \ldots, F_6$

procedure `Query`(i,U)
  if $U \notin F_i$ **then**
    $F_i(U) \leftarrow_{\mathcal{R}} \{0,1\}^n$
    if $i = 3$ **then**
      for all $Z \in F_4$ **do**
        `CompleteBackward`(U, Z)
      end for
    end if
    if $i = 4$ **then**
      for all $Y \in F_3$ **do**
        `CompleteForward`(Y, U)
      end for
    end if
  end if
  return $F_i(U)$
end procedure

procedure `CompleteForward`(Y,Z)
  $X := Z \oplus F_3(Y)$
  `Query`$(2, X)$
  $R := Y \oplus F_2(X)$
  `Query`$(1, R)$
  $L := X \oplus F_1(R)$
  $(S,T) := \boldsymbol{P}(0,(L,R))$ $\boxed{(S,T) := \boldsymbol{R}(0,(L,R))}$
  $A := Y \oplus F_4(Z)$
  $F_5(A) \leftarrow Z \oplus S$
  $F_6(S) \leftarrow A \oplus T$
end procedure

procedure `CompleteBackward`(Y,Z)
  $A := Y \oplus F_4(Z)$
  `Query`$(5, A)$
  $S := Z \oplus F_5(A)$
  `Query`$(6, S)$
  $T := A \oplus F_6(S)$
  $(L,R) := \boldsymbol{P}(1,(S,T))$ $\boxed{(L,R) := \boldsymbol{R}(1,(S,T))}$
  $X := Z \oplus F_3(Y)$
  $F_2(X) \leftarrow R \oplus Y$
  $F_1(R) \leftarrow L \oplus X$
end procedure

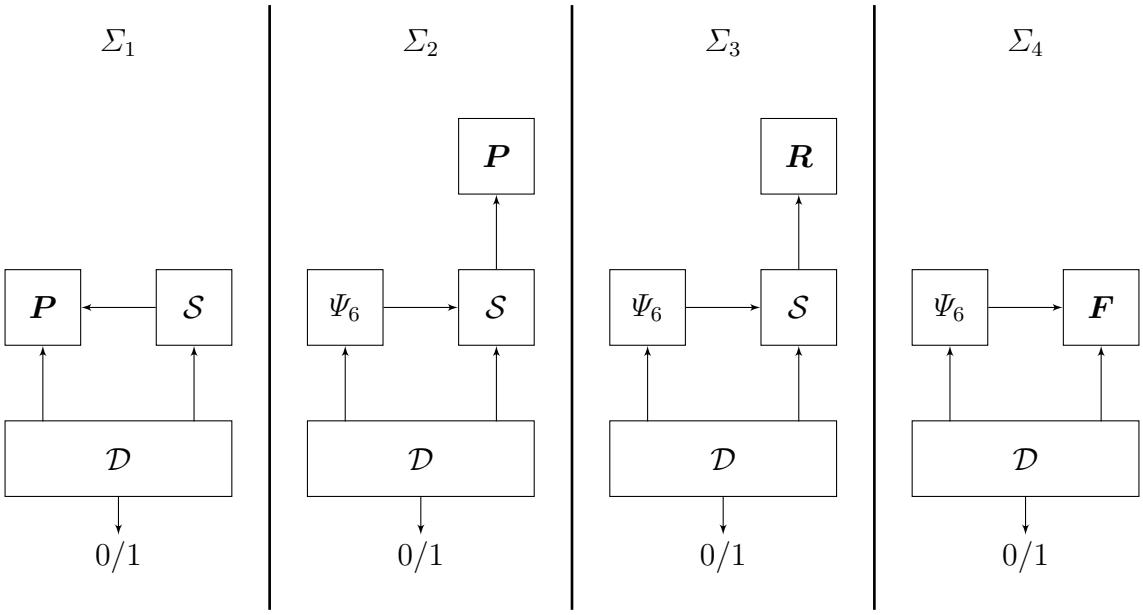

Fig. 3. Systems used in the seq-indifferentiability proof.

Lemma 1. *When the simulator is asked at most q queries, then the size of histories for F_3 and F_4 is at most q, the size of histories for F_1, F_2, F_5 and F_6 is at most q^2+q, the procedures* `CompleteForward` *and* `CompleteBackward` *are called in total at most q^2 times, and the simulator makes at most q^2 queries to the random permutation.*

Proof. Elements are added to the history of F_3 and F_4 only when a corresponding F-query is made to the simulator, so that the size of their history cannot be greater than q. For each pair $(Y, Z) \in F_3 \times F_4$, either `CompleteForward`$(Y, Z)$ or `CompleteBackward`(Y, Z) is called, at most once, so that in total these procedures are called at most q^2 times. Since the simulator makes one query to the random permutation per execution of `CompleteForward` and `CompleteBackward` this in turns implies that the total number of queries to $\boldsymbol{P}$ is at most q^2. Finally, elements are added to the history of F_1, F_2, F_5 and F_6 either when a query is made to the simulator, or during an execution of `CompleteForward` or `CompleteBackward`, so that the size of their history cannot be greater than $q^2 + q$. □

In order to prove that the two systems $\Sigma_1 = (\boldsymbol{P}, \mathcal{S}^{\boldsymbol{P}})$ and $\Sigma_4 = (\Psi_6^{\boldsymbol{F}}, \boldsymbol{F})$ are indistinguishable, we will use two intermediate systems: $\Sigma_2 = (\Psi_6^{\boldsymbol{S}^{\boldsymbol{P}}}, \boldsymbol{S}^{\boldsymbol{P}})$ where the P-queries of $\mathcal{D}$ are answered by the Feistel construction asking round function values to the simulator, which itself interacts with $\boldsymbol{P}$, and $\Sigma_3 = (\Psi_6^{\boldsymbol{S}^{\boldsymbol{R}}}, \boldsymbol{S}^{\boldsymbol{R}})$ where the random invertible permutation is replaced by a two-sided random function $\boldsymbol{R}$ (note the corresponding change in procedures `CompleteForward` and `CompleteBackward` indicated by a boxed statement). The four systems used in the proof are depicted in Figure 3.

The main part of the analysis is concerned with systems Σ_2 and Σ_3. We will show that unless some bad event happens, the round function values set by the simulator in Σ_2 are consistent with $\boldsymbol{P}$ (which will enable to bound the statistical distance between Σ_1 and Σ_2), and that in Σ_3 they are uniformly random and independent (which will enable to bound the statistical distance between Σ_3 and Σ_4). In systems Σ_2 and Σ_3, the simulator first receives at most q_f queries from the distinguisher, and then at most $6q_p$ queries from the Feistel construction (6 for each P-query of the distinguisher). Hence the total number of queries received by the simulator is exactly the total oracle queries cost of $\mathcal{D}$, which is less than q. The statistical distance between answers of systems Σ_2 and Σ_3 is easily bounded.

Lemma 2. *For any distinguisher of total oracle queries cost at most q, the following holds:*

$$\left|\Pr\left[\mathcal{D}^{\Sigma_2}(1^n)=1\right]-\Pr\left[\mathcal{D}^{\Sigma_3}(1^n)=1\right]\right| \leq \frac{q^4}{2^{2n+1}} \ .$$

Proof. Consider the union of $\mathcal{D}$, Ψ_6, and $\mathcal{S}$ as a single distinguisher $\mathcal{D}'$ interacting either with a random invertible permutation or a two-sided random function. Note that $\mathcal{D}'$ makes at most q^2 queries to its oracle (Lemma 1). One can conclude thanks to the PRF/PRP switching lemma [BR06]. □

Before going further with the proof, we define formally what it means for an input $x \in \{0,1\}\times\{0,1\}^n$ to the Feistel construction to be computable with respect to the history of the simulator.

Definition 4 (Computable input). *Given a simulator history $\mathcal{H}$ and an input $x \in \{0,1\}\times\{0,1\}^{2n}$, the sequence $\rho_{\mathcal{H}}(x) = (\rho_{\mathcal{H}}(x)[i])_{i\in[0..7]}$ is defined as follows:*

- *for a forward input $x = (0,(L,R))$, $\rho_{\mathcal{H}}(x)[0] = L$, $\rho_{\mathcal{H}}(x)[1] = R$, and for $i = 2$ to 7:*

$$\begin{cases} \text{if } \rho_{\mathcal{H}}(x)[i-1] \in F_{i-1} \text{ then } \rho_{\mathcal{H}}(x)[i] = \rho_{\mathcal{H}}(x)[i-2] \oplus F_{i-1}(\rho_{\mathcal{H}}(x)[i-1]) \\ \text{else } \rho_{\mathcal{H}}(x)[i] = \perp \end{cases}$$

- *for a backward input $x = (1,(S,T))$, $\rho_{\mathcal{H}}(x)[7] = T$, $\rho_{\mathcal{H}}(x)[6] = S$, and for $i = 5$ to 0:*

$$\begin{cases} \text{if } \rho_{\mathcal{H}}(x)[i+1] \in F_{i+1} \text{ then } \rho_{\mathcal{H}}(x)[i] = \rho_{\mathcal{H}}(x)[i+2] \oplus F_{i+1}(\rho_{\mathcal{H}}(x)[i+1]) \\ \text{else } \rho_{\mathcal{H}}(x)[i] = \perp \end{cases}$$

An input x is said to be computable with respect to $\mathcal{H}$ *iff $\rho_{\mathcal{H}}(x)[i] \neq \perp$ for all $i \in [0..7]$. In that case we note $\Psi_6^{\mathcal{H}}(x) = (\rho_{\mathcal{H}}(x)[6], \rho_{\mathcal{H}}(x)[7])$ if x is a forward input and $\Psi_6^{\mathcal{H}}(x) = (\rho_{\mathcal{H}}(x)[0], \rho_{\mathcal{H}}(x)[1])$ if x is a backward input.*

For a computable input x, we will often use the notation $(L,R,X,Y,Z,A,S,T) = \rho_{\mathcal{H}}(x)$ as depicted on Figure 1.

We now define a bad event that may occur during the execution of the simulator (in Σ_2 or Σ_3) in relation with Lines 26, 27, 37, and 38 of the simulator. We will say that event `Bad` happens if in any execution of `CompleteForward` or `CompleteBackward`, the input value whose image is set at Lines 26, 27, 37 or 38 is already in the history of the corresponding round function. This implies that the simulator overwrites a value so that its answers may not be coherent with $\boldsymbol{P}$ or $\boldsymbol{R}$ any more.[10] Reciprocally, if `Bad` does not happen, then the simulator never overwrites any value in its history.

We start with the simple observation that if `Bad` does not happen, then during any execution of `CompleteForward` or `CompleteBackward`, the query to $\boldsymbol{P}$ or $\boldsymbol{R}$ made by the simulator is fresh.

Lemma 3. *In system Σ_2, if* `Bad` *does not happen, then in any execution of* `CompleteForward` *or* `CompleteBackward` *the query to $\boldsymbol{P}$ made by the simulator is not in the history of $\boldsymbol{P}$. For system Σ_3, the corresponding statement holds for $\boldsymbol{R}$.*

Proof. The reasoning is the same for Σ_2 and Σ_3, we use Σ_2 to fix ideas. Consider an execution of `CompleteForward`(Y,Z). Let $x = (0,(L,R))$ be the query to $\boldsymbol{P}$ made by the simulator, and $(S,T) = \boldsymbol{P}(x)$. If x is already in the history of $\boldsymbol{P}$, then it was necessarily added by a previous execution of `CompleteForward`(Y',Z') or `CompleteBackward`(Y',Z') (note that the distinguisher does not make any query to $\boldsymbol{P}$ in Σ_2 or to $\boldsymbol{R}$ in Σ_3). But since `Bad` does not happen, round function values are never overwritten so that necessarily $(Y',Z') = (Y,Z)$. This is impossible since by construction the simulator makes at most one call to `CompleteForward` or `CompleteBackward` per center $(Y,Z) \in F_3 \times F_4$. □

We are now ready to upper bound the probability that `Bad` happens in Σ_2 or Σ_3.

[10] In previous work on indifferentiability of the Feistel construction [CPS08,Seu09], in such a case the simulator aborted. It does not change much since, as we will prove, this happens only with negligible probability.

Lemma 4. *For any distinguisher of total oracle queries cost at most q, event* `Bad` *happens with probability less than $4q^4/2^n$ in Σ_3 and less than $4q^4/2^n + q^4/2^{2n+1}$ in Σ_2.*

Proof. We start by working with Σ_3 since it is slightly easier to analyze. Assume `Bad` has not happened yet, and consider a call to $\texttt{Query}(3, Y)$ (the case of $\texttt{Query}(4, Z)$ is symmetric). Let $Z_1, \ldots, Z_m$, ($m \leq q$ according to Lemma 1) be the values in the history of F_4 at this point. We show that event `Bad` does not happen for any call to $\texttt{CompleteBackward}(Y, Z_i)$ except with negligible probability. Since $F_3(Y)$ is set uniformly at random, the probability that any value $X_i = Z_i \oplus F_3(X)$ is in the history of F_2 at the time $F_3(Y)$ is set is less than $m(q^2+q)/2^n \leq 2q^3/2^n$. Moreover $F_2(X_i)$ cannot be set until $\texttt{CompleteBackward}(Y, Z_i)$ is called, hence `Bad` does not happen for Line 37 of any execution of $\texttt{CompleteBackward}(Y, Z_i)$ except with probability less than $2q^3/2^n$. Let (L_i, R_i) denote the answer of the query to $\boldsymbol{R}$ in $\texttt{CompleteBackward}(Y, Z_i)$. Since `Bad` has not happened yet, according to Lemma 3 this query is not in the history of $\boldsymbol{R}$ so that the answer is uniformly random. Hence R_i is in the history of F_1 with probability less than $(q^2+q)/2^n$. Since there are $m \leq q$ calls to `CompleteBackward`, event `Bad` does not occur for Line 38 of `CompleteBackward` except with probability less than $2q^3/2^n$. Finally, since there are at most q calls in total to $\texttt{Query}(3, \cdot)$ and $\texttt{Query}(4, \cdot)$, event `Bad` happens with probability less than $4q^4/2^n$ in Σ_3. The absolute difference between the probability that `Bad` happens in Σ_2 and Σ_3 cannot be greater than the statistical distance between answers of $\boldsymbol{P}$ and $\boldsymbol{R}$, hence the probability that `Bad` happens in Σ_2 is less than $4q^4/2^n + q^4/2^{2n+1}$. □

The following lemma says that as long as `Bad` does not happen in Σ_2, the round function values set by the simulator are consistent with $\boldsymbol{P}$.

Lemma 5. *If* `Bad` *does not happen in system Σ_2, then for any input $x \in \{0,1\} \times \{0,1\}^{2n}$ computable with respect to the final history of the simulator $\mathcal{H}$, $\Psi_6^{\mathcal{H}}(x) = \boldsymbol{P}(x)$.*

Proof. Consider an input $x \in \{0,1\} \times \{0,1\}^{2n}$ computable with respect to the final history $\mathcal{H}$ of the simulator, and let $(L, R, X, Y, Z, A, S, T) = \rho_{\mathcal{H}}(x)$. There was necessarily a call to $\texttt{CompleteForward}(Y, Z)$ or $\texttt{CompleteBackward}(Y, Z)$ during the execution of the simulator. With respect to the history $\mathcal{H}'$ just after the completion of $\texttt{CompleteForward}(Y, Z)$ or $\texttt{CompleteBackward}(Y, Z)$, it is clear that $\Psi_6^{\mathcal{H}'}(x) = \boldsymbol{P}(x)$. Since `Bad` does not happen the simulator never overwrites a value and the equality remains true until the end of the simulation, hence $\Psi_6^{\mathcal{H}}(x) = \boldsymbol{P}(x)$. □

A direct consequence of this lemma is that as long as `Bad` does not happen in Σ_2, the answers of systems Σ_1 and Σ_2 are identically distributed.

Lemma 6. *For any distinguisher of total oracle queries cost at most q, the following holds:*

$$\left|\Pr\left[\mathcal{D}^{\Sigma_1}(1^n) = 1\right] - \Pr\left[\mathcal{D}^{\Sigma_2}(1^n) = 1\right]\right| \leq \frac{4q^4}{2^n} + \frac{q^4}{2^{2n+1}} \ .$$

Proof. Clearly, answers to F-queries of the distinguisher are identically distributed in Σ_1 and Σ_2 since they are answered by $\mathcal{S}^{\boldsymbol{P}}$ in both systems (may `Bad` occur or not).[11] Moreover, in Σ_2 any P-query x asked by the distinguisher is computable with respect to the history of the simulator at the time it is answered by Ψ_6, and if `Bad` does not happen in Σ_2, then according to Lemma 5, $\Psi_6^{\mathcal{H}}(x) = \boldsymbol{P}(x)$ so that answers to P-queries of the distinguisher are also identically distributed in both systems. The result follows from Lemma 4. □

Lemma 7. *If* `Bad` *does not happen in system Σ_3, then the round function values set by the simulator are uniformly random and independent.*

[11] It is crucial here that the distinguisher is sequential, otherwise the simulation in Σ_2 would be altered by the queries made by Ψ_6.

Proof. Since this is clear for round function values set uniformly at random (independently of `Bad` occurring or not), we only have to examine values that are adapted at Lines 26, 27, 37, and 38 of the simulator. But according to Lemma 3, if `Bad` does not happen, the query to $\boldsymbol{R}$ made by the distinguisher in any execution of `CompleteForward` or `CompleteBackward` is not in the history of $\boldsymbol{R}$, so that the answer (S, T) or (L, R) is uniformly random. Consequently, round function values set by $F_5(A) \leftarrow Z \oplus S$ and $F_6(S) \leftarrow A \oplus T$ in `CompleteForward`, or $F_2(X) \leftarrow R \oplus Y$ and $F_1(R) \leftarrow L \oplus X$ in `CompleteBackward` are uniformly random and independent of previous round function values set by the simulator. Since `Bad` does not happen round function values are not overwritten and the result follows. □

This lemma finally enables to bound the statistical distance between the answers of Σ_3 and Σ_4.

Lemma 8. *For any distinguisher of total oracle queries cost at most q, the following holds:*

$$\left|\Pr\left[\mathcal{D}^{\Sigma_3}(1^n) = 1\right] - \Pr\left[\mathcal{D}^{\Sigma_4}(1^n) = 1\right]\right| \leq \frac{4q^4}{2^n} \ .$$

Proof. If `Bad` does not occur in Σ_3 then answers of $\mathcal{S}^{\boldsymbol{R}}$ are distributed exactly as answers of $\boldsymbol{F}$ according to Lemma 7. Hence the statistical distance between answers of Σ_3 and Σ_4 is upper bounded by the probability that `Bad` happens in Σ_3, given by Lemma 4. □

Theorem 2 is now a simple consequence of Lemmata 2, 6, and 8.

Remark 1. The strategy of using the intermediate system Σ_2 is likely to be quite generic for seq-indifferentiability proofs (system Σ_3, on the contrary, is quite specific to the Feistel construction). We believe this could probably make proofs of pub-indifferentiability (*e.g.* [DRS09, Section 7]) much easier, but leave this for future work.

Remark 2. Note that for general distinguishers (not necessarily sequential), the proof would go through exactly as above for Lemmata 2 and 8. The problematic step is clearly going from Σ_1 to Σ_2. To see what could go wrong if the distinguisher can interleave queries to $\boldsymbol{P}$ and $\mathcal{S}$, consider the following simple example. $\mathcal{D}$ first makes a P-query $\boldsymbol{P}(0, (L, R)) = (S, T)$, and then makes the sequence of F-queries $F_1(R)$, $F_2(X)$, $F_6(S)$, $F_5(A)$. In system Σ_1, the simulator returns uniformly answers to the four F-queries and will be unable to adapt F_3 and F_4, whereas in Σ_2 the initial P-query of the distinguisher will trigger six F-queries from Ψ_6 which will lead the simulator to adapt the chain when query $F_4(Y)$ occurs. Making progress towards proving full indifferentiability for six rounds clearly requires to find the right way to deal with these "external" chains without knowing the P-queries of the distinguisher.

4 Applications to Correlation Intractability

Correlation intractability was introduced by Canetti *et al.* in their work on the limits of the random oracle methodology [CGH98]. In the standard model, a function family is said to be correlation intractable if given the description of a random function f of the family, no PPT algorithm can find an input x, or more generally a sequence of inputs $(x_1, \ldots, x_m)$, such that $((x_1, \ldots, x_m), (f(x_1), \ldots, f(x_m)))$ satisfies a relation that would be hard to satisfy for a uniformly random function.

There is no difficulty in extending the definition of correlation intractability to an idealized model: instead of passing the description of the function as input to the algorithm, it is granted access to the ideal primitive used by the construction $\mathcal{C}$. This way one can define a correlation intractable construction (accessing an ideal primitive).

In all the following, we will consider relations over pairs of binary sequences (formally, a subset of $\{0,1\}^* \times \{0,1\}^*$). We assume that the machine $\mathcal{M}$ returns sequences of strings in $\mathtt{Dom}_n$, the domain of the ideal primitive $\boldsymbol{G}_n$ or the construction $\mathcal{C}^{\boldsymbol{F}_n}$.

Definition 5 (Evasive relation). *Let $\boldsymbol{G} = (\boldsymbol{G}_n)$ be an ideal primitive associated to $\mathbb{G} = (\mathtt{Dom}_n, \mathtt{Rng}_n, \mathbb{G}_n)$. A relation $\mathcal{R}$ over pairs of binary sequences is said to be evasive with respect to $\boldsymbol{G}$ if for any PPT oracle machine $\mathcal{M}$, there is a negligible function ϵ such that the following holds:*

$$\Pr\left[(x_1, \ldots, x_m) \leftarrow \mathcal{M}^{\boldsymbol{G}_n}(1^n) \,:\, ((x_1, \ldots, x_m), (\boldsymbol{G}_n(x_1), \ldots, \boldsymbol{G}_n(x_m))) \in \mathcal{R}\right] \leq \epsilon(n) \ .$$

Example 1. The relation over pairs of quadruplets of binary strings

$$\cup_n \left\{ \left(((0,(L_1,R_1)),(0,(L_2,R_2)),(0,(L_3,R_3)),(0,(L_4,R_4))), ((S_1,T_1),(S_2,T_2),(S_3,T_3),(S_4,T_4)) \right) : \right.$$
$$\left. L_i, R_i, S_i, T_i \in \{0,1\}^n \text{ and } R_1 \oplus R_2 \oplus R_3 \oplus R_4 = 0 \text{ and } S_1 \oplus S_2 \oplus S_3 \oplus S_4 = 0 \right\}$$

is evasive for a random invertible permutation. This is exactly the evasive relation used in Appendix C to show that the 5-round Feistel construction is not seq-indifferentiable from a random permutation. This same attack also shows that the 5-round Feistel construction is not correlation intractable.

Definition 6 (Correlation intractable construction). *Let $\mathcal{C}$ be a construction with oracle access to an ideal primitive $\boldsymbol{F} = (\boldsymbol{F}_n)$ and implementing some primitive $\mathbb{G}$. $\mathcal{C}^{\boldsymbol{F}}$ is said to be (multiple-output) correlation intractable if for any relation $\mathcal{R}$ over pairs of binary sequences evasive with respect to $\boldsymbol{G}$, and any PPT oracle machine $\mathcal{M}$, there is a negligible function ϵ such that:*

$$\Pr\left[(x_1, \ldots, x_m) \leftarrow \mathcal{M}^{\boldsymbol{F}_n}(1^n) \,:\, \left((x_1, \ldots, x_m), (\mathcal{C}^{\boldsymbol{F}_n}(x_1), \ldots, \mathcal{C}^{\boldsymbol{F}_n}(x_m))\right) \in \mathcal{R}\right] \leq \epsilon(n) \ .$$

Weak correlation intractability is defined similarly as above by quantifying only over all polynomial-time recognizable relations (*i.e.* relations $\mathcal{R}$ such that there exists a polynomial-time algorithm that, given $((x_1, \ldots, x_m), (y_1, \ldots, y_m))$, decides whether it belongs to $\mathcal{R}$ or not).

Theorem 3. *Let $\mathcal{C}$ be a construction with oracle access to an ideal primitive $\boldsymbol{F} = (\boldsymbol{F}_n)$ and implementing some primitive $\mathbb{G}$. If $\mathcal{C}^{\boldsymbol{F}}$ is statistically (resp. computationally) seq-indifferentiable from the ideal primitive $\boldsymbol{G}$, then $\mathcal{C}^{\boldsymbol{F}}$ is correlation intractable (resp. weakly correlation intractable).*

Proof. Assume that $\mathcal{C}^{\boldsymbol{F}}$ is not correlation intractable. Then there is an evasive relation $\mathcal{R}$ and a PPT oracle machine $\mathcal{M}$ such that $\mathcal{M}^{\boldsymbol{F}_n}(1^n)$ outputs with non-negligible probability δ a sequence $(x_1, \ldots, x_m)$ such that $((x_1, \ldots, x_m), (\mathcal{C}^{\boldsymbol{F}_n}(x_1), \ldots, \mathcal{C}^{\boldsymbol{F}_n}(x_m))) \in \mathcal{R}$. Consider the following sequential distinguisher $\mathcal{D}$ accessing a pair of oracles (G, F): it runs $\mathcal{M}$, answering $\mathcal{M}$'s oracle queries with its own oracle F. $\mathcal{M}$ returns $(x_1, \ldots, x_m)$. $\mathcal{D}$ then makes oracle queries $G(x_1), \ldots, G(x_m)$ and checks[12] whether $((x_1, \ldots, x_m), (G(x_1), \ldots, G(x_m))) \in \mathcal{R}$. If this is the case it returns 1, otherwise it returns 0.

When the distinguisher is interacting with $(\mathcal{C}^{\boldsymbol{F}}, \boldsymbol{F})$, the probability that it returns 1 is exactly δ, which is non-negligible by hypothesis. On the contrary, when it interacts with $(\boldsymbol{G}, \mathcal{S}^{\boldsymbol{G}})$, then the union of $\mathcal{M}$ and $\mathcal{S}$ is a PPT oracle machine with oracle access to $\boldsymbol{G}$, so that by definition of an evasive relation $\mathcal{D}$ outputs 1 only with negligible probability. The advantage of the distinguisher is non-negligible, which contradicts the seq-indifferentiability of $\mathcal{C}^{\boldsymbol{F}}$. □

A direct consequence of Theorems 2 and 3 is that the 6-round Feistel construction with random round functions is correlation intractable: no polynomial algorithm with oracle access to the round functions can find a sequence of inputs that together with their image by the Feistel satisfy a relation that would be hard to satisfy in the random invertible permutation model. Note that the sole *existence* of correlation intractable invertible permutations in the random oracle model was already implied by the result of Holenstein *et al.* [HKT11] on the full indifferentiability of the 14-round Feistel construction (since full indifferentiability implies seq-indifferentiability and hence correlation intractability), but our results shows that six rounds are sufficient to achieve this property.

[12] Note that the reasoning holds only relatively to a polynomial-time recognizable relation if $\mathcal{D}$ is computationally bounded.

Remark 3. According to Theorem 3, sequential indifferentiability implies correlation intractability. However correlation intractability does not necessarily imply sequential indifferentiability. In Appendix D we provide a simple counter-example separating the two notions.

Implications for Chosen-Key and Known-Key Attacks on Block Ciphers. Knudsen and Rijmen [KR07] have introduced so-called known-key attacks on block ciphers. We discuss the implications of our results regarding this attack model in Appendix E.

References

[BDPA08] Guido Bertoni, Joan Daemen, Michael Peeters, and Gilles Van Assche. On the Indifferentiability of the Sponge Construction. In Nigel P. Smart, editor, *Advances in Cryptology - EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 181–197. Springer, 2008.

[BR93] Mihir Bellare and Phillip Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73, 1993.

[BR96] Mihir Bellare and Phillip Rogaway. The Exact Security of Digital Signatures - How to Sign with RSA and Rabin. In Ueli M. Maurer, editor, *Advances in Cryptology - EUROCRYPT '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 399–416. Springer, 1996.

[BR06] Mihir Bellare and Thomas Ristenpart. Multi-Property-Preserving Hash Domain Extension and the EMD Transform. In Xuejia Lai and Kefei Chen, editors, *Advances in Cryptology - ASIACRYPT 2006*, volume 4284 of *Lecture Notes in Computer Science*, pages 299–314. Springer, 2006.

[BRS02] John Black, Phillip Rogaway, and Thomas Shrimpton. Black-Box Analysis of the Block-Cipher-Based Hash-Function Constructions from PGV. In Moti Yung, editor, *Advances in Cryptology - CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 320–335. Springer, 2002.

[CDMP05] Jean-Sébastien Coron, Yevgeniy Dodis, Cécile Malinaud, and Prashant Puniya. Merkle-Damgård Revisited: How to Construct a Hash Function. In Victor Shoup, editor, *Advances in Cryptology - CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 430–448. Springer, 2005.

[CDMS10] Jean-Sébastien Coron, Yevgeniy Dodis, Avradip Mandal, and Yannick Seurin. A Domain Extender for the Ideal Cipher. In Daniele Micciancio, editor, *Theory of Cryptography Conference - TCC 2010*, volume 5978 of *Lecture Notes in Computer Science*, pages 273–289. Springer, 2010.

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The Random Oracle Methodology, Revisited (Preliminary Version). In *Symposium on Theory of Computing - STOC '98*, pages 209–218. ACM, 1998. Full version available at http://arxiv.org/abs/cs.CR/0010019.

[CLNY06] Donghoon Chang, Sangjin Lee, Mridul Nandi, and Moti Yung. Indifferentiable Security Analysis of Popular Hash Functions with Prefix-Free Padding. In Xuejia Lai and Kefei Chen, editors, *Advances in Cryptology - ASIACRYPT 2006*, volume 4284 of *Lecture Notes in Computer Science*, pages 283–298. Springer, 2006.

[CMPP05] Benoît Chevallier-Mames, Duong Hieu Phan, and David Pointcheval. Optimal Asymmetric Encryption and Signature Paddings. In John Ioannidis, Angelos D. Keromytis, and Moti Yung, editors, *Applied Cryptography and Network Security - ACNS 2005*, volume 3531 of *Lecture Notes in Computer Science*, pages 254–268. Springer, 2005.

[CN08] Donghoon Chang and Mridul Nandi. Improved Indifferentiability Security Analysis of chopMD Hash Function. In Kaisa Nyberg, editor, *Fast Software Encryption - FSE 2008*, volume 5086 of *Lecture Notes in Computer Science*, pages 429–443. Springer, 2008.

[Cor02] Jean-Sébastien Coron. Optimal Security Proofs for PSS and Other Signature Schemes. In Lars R. Knudsen, editor, *Advances in Cryptology - EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 272–287. Springer, 2002.

[CPS08] Jean-Sébastien Coron, Jacques Patarin, and Yannick Seurin. The Random Oracle Model and the Ideal Cipher Model Are Equivalent. In David Wagner, editor, *Advances in Cryptology - CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2008.

[Dam89] Ivan Damgård. A Design Principle for Hash Functions. In Gilles Brassard, editor, *Advances in Cryptology - CRYPTO '89*, volume 435 of *Lecture Notes in Computer Science*, pages 416–427. Springer, 1989.

[DP06] Yevgeniy Dodis and Prashant Puniya. On the Relation Between the Ideal Cipher and the Random Oracle Models. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography Conference - TCC 2006*, volume 3876 of *Lecture Notes in Computer Science*, pages 184–206. Springer, 2006.

[DP07] Yevgeniy Dodis and Prashant Puniya. Feistel Networks Made Public, and Applications. In Moni Naor, editor, *Advances in Cryptology - EUROCRYPT 2007*, volume 4515 of *Lecture Notes in Computer Science*, pages 534–554. Springer, 2007.

[DRRS09] Yevgeniy Dodis, Leonid Reyzin, Ronald L. Rivest, and Emily Shen. Indifferentiability of Permutation-Based Compression Functions and Tree-Based Modes of Operation, with Applications to MD6. In Orr Dunkelman, editor, *Fast Software Encryption - FSE 2009*, volume 5665 of *Lecture Notes in Computer Science*, pages 104–121. Springer, 2009.

[DRS09] Yevgeniy Dodis, Thomas Ristenpart, and Thomas Shrimpton. Salvaging Merkle-Damgård for Practical Applications. In Antoine Joux, editor, *Advances in Cryptology - EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 371–388. Springer, 2009.

[FLP08] Marc Fischlin, Anja Lehmann, and Krzysztof Pietrzak. Robust Multi-property Combiners for Hash Functions Revisited. In Ivan Damgård, editor, *International Colloquium on Automata, Languages and Programming - ICALP 2008, Track C: Security and Cryptography Foundations*, volume 5126 of *Lecture Notes in Computer Science*, pages 655–666. Springer, 2008.

[GR04] Craig Gentry and Zulfikar Ramzan. Eliminating Random Permutation Oracles in the Even-Mansour Cipher. In Pil Joong Lee, editor, *Advances in Cryptology - ASIACRYPT 2004*, volume 3329 of *Lecture Notes in Computer Science*, pages 32–47. Springer, 2004.

[Gra02] Louis Granboulan. Short Signatures in the Random Oracle Model. In Yuliang Zheng, editor, *Advances in Cryptology - ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 364–378. Springer, 2002.

[HKT11] Thomas Holenstein, Robin Künzler, and Stefano Tessaro. The Equivalence of the Random Oracle Model and the Ideal Cipher Model, Revisited. In *STOC 2011*, 2011. To appear.

[HPY07] Shoichi Hirose, Je Hong Park, and Aaram Yun. A Simple Variant of the Merkle-Damgård Scheme with a Permutation. In Kaoru Kurosawa, editor, *Advances in Cryptology - ASIACRYPT 2007*, volume 4833 of *Lecture Notes in Computer Science*, pages 113–129. Springer, 2007.

[KR07] Lars R. Knudsen and Vincent Rijmen. Known-Key Distinguishers for Some Block Ciphers. In Kaoru Kurosawa, editor, *Advances in Cryptology - ASIACRYPT 2007*, volume 4833 of *Lecture Notes in Computer Science*, pages 315–324. Springer, 2007.

[Kün09] Robin Künzler. Are the random oracle and the ideal cipher models equivalent? Master's thesis, ETH Zurich, Switzerland, 2009.

[KW03] Jonathan Katz and Nan Wang. Efficiency improvements for signature schemes with tight security reductions. In Sushil Jajodia, Vijayalakshmi Atluri, and Trent Jaeger, editors, *ACM Conference on Computer and Communications Security*, pages 155–164. ACM, 2003.

[LR88] Michael Luby and Charles Rackoff. How to Construct Pseudorandom Permutations from Pseudorandom Functions. *SIAM Journal on Computing*, 17(2):373–386, 1988.

[Mau92] Ueli M. Maurer. A Simplified and Generalized Treatment of Luby-Rackoff Pseudorandom Permutation Generator. In Rainer A. Rueppel, editor, *Advances in Cryptology - EUROCRYPT '92*, volume 658 of *Lecture Notes in Computer Science*, pages 239–255. Springer, 1992.

[Mau02] Ueli M. Maurer. Indistinguishability of Random Systems. In Lars R. Knudsen, editor, *Advances in Cryptology - EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 110–132. Springer, 2002.

[Mer89] Ralph C. Merkle. One Way Hash Functions and DES. In Gilles Brassard, editor, *Advances in Cryptology - CRYPTO '89*, volume 435 of *Lecture Notes in Computer Science*, pages 428–446. Springer, 1989.

[MRH04] Ueli M. Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, Impossibility Results on Reductions, and Applications to the Random Oracle Methodology. In Moni Naor, editor, *Theory of Cryptography Conference- TCC 2004*, volume 2951 of *Lecture Notes in Computer Science*, pages 21–39. Springer, 2004.

[MT07] Ueli M. Maurer and Stefano Tessaro. Domain Extension of Public Random Functions: Beyond the Birthday Barrier. In Alfred Menezes, editor, *Advances in Cryptology - CRYPTO 2007*, volume 4622 of *Lecture Notes in Computer Science*, pages 187–204. Springer, 2007.

[NR99] Moni Naor and Omer Reingold. On the Construction of Pseudorandom Permutations: Luby-Rackoff Revisited. *Journal of Cryptology*, 12(1):29–66, 1999.

[Pat90] Jacques Patarin. Pseudorandom Permutations Based on the DES Scheme. In Gérard D. Cohen and Pascale Charpin, editors, *EUROCODE '90*, volume 514 of *Lecture Notes in Computer Science*, pages 193–204. Springer, 1990.

[Pat91] Jacques Patarin. New Results on Pseudorandom Permutation Generators Based on the DES Scheme. In Joan Feigenbaum, editor, *Advances in Cryptology - CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*, pages 301–312. Springer, 1991.

[Pat98] Jacques Patarin. About Feistel Schemes with Six (or More) Rounds. In Serge Vaudenay, editor, *Fast Software Encryption - FSE '98*, volume 1372 of *Lecture Notes in Computer Science*, pages 103–121. Springer, 1998.

[Pat03] Jacques Patarin. Luby-Rackoff: 7 Rounds Are Enough for $2^{n(1-\epsilon)}$ Security. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 513–529. Springer, 2003.

[Pat04] Jacques Patarin. Security of Random Feistel Schemes with 5 or More Rounds. In Matthew K. Franklin, editor, *Advances in Cryptology - CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 106–122. Springer, 2004.

[PGV93] Bart Preneel, René Govaerts, and Joos Vandewalle. Hash Functions Based on Block Ciphers: A Synthetic Approach. In Douglas R. Stinson, editor, *Advances in Cryptology - CRYPTO '93*, volume 773 of *Lecture Notes in Computer Science*, pages 368–378. Springer, 1993.

[RR00] Zulfikar Ramzan and Leonid Reyzin. On the Round Security of Symmetric-Key Cryptographic Primitives. In Mihir Bellare, editor, *Advances in Cryptology - CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 376–393. Springer, 2000.

[RSS11] Thomas Ristenpart, Hovav Shacham, and Thomas Shrimpton. Careful with Composition: Limitations of the Indifferentiability Framework. In Kenneth G. Paterson, editor, *Advances in Cryptology - EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 487–506. Springer, 2011.

[Seu09] Yannick Seurin. *Primitives et protocoles cryptographiques à sécurité prouvée*. PhD thesis, Université de Versailles Saint-Quentin-en-Yvelines, France, 2009.

[Vau03] Serge Vaudenay. Decorrelation: A Theory for Block Cipher Security. *Journal of Cryptology*, 16(4):249–286, 2003.

[YMO09] Kazuki Yoneyama, Satoshi Miyagawa, and Kazuo Ohta. Leaky Random Oracle. *IEICE Transactions*, 92-A(8):1795–1807, 2009.

A Proof of Theorem 1

Assume that $\mathcal{C}^{\boldsymbol{F}}$ is strongly $(2q, \sigma, \epsilon)$-seq-indifferentiable from $\boldsymbol{G}$, and let $\mathcal{S}_{\text{seq}}$ be the simulator for seq-indifferentiability. We define a simulator $\mathcal{S}_{\text{pub}}$ for pub-indifferentiability as follows. $\mathcal{S}_{\text{pub}}$ runs $\mathcal{S}_{\text{seq}}$, transparently relaying queries of $\mathcal{S}_{\text{seq}}$ to $\boldsymbol{G}$ to the regular interface of $\overline{\boldsymbol{G}}$. Each time $\mathcal{S}_{\text{pub}}$ receives a F-query y from the distinguisher, it makes a call to `Reveal`, getting a sequence $(x_1, \ldots, x_m)$ of G-queries that have been made by the distinguisher so far ($\mathcal{S}_{\text{pub}}$ considers only *fresh* G-queries, *i.e.* G-queries that have not been returned by a previous query to `Reveal`). For each $i = 1$ to m, $\mathcal{S}_{\text{pub}}$ makes all F-queries needed to compute $\mathcal{C}^F(x_i)$ to $\mathcal{S}_{\text{seq}}$. Finally, it makes the F-query y to $\mathcal{S}_{\text{seq}}$ and returns the corresponding answer.

Let $\mathcal{D}_{\text{pub}}$ be a distinguisher for the pub-indifferentiability game of total oracle queries cost at most q. We have to bound the absolute difference between the probabilities that $\mathcal{D}_{\text{pub}}$ outputs 1 when interacting with $(\boldsymbol{G}, \mathcal{S}_{\text{pub}}^{\overline{\boldsymbol{G}}})$ and $(\mathcal{C}^{\boldsymbol{F}}, \boldsymbol{F})$. For this, we assume *wlog* that

$$\Pr\left[\mathcal{D}_{\text{pub}}^{\boldsymbol{G}, \mathcal{S}_{\text{pub}}^{\overline{\boldsymbol{G}}}}(1^n) = 1\right] \geq \Pr\left[\mathcal{D}_{\text{pub}}^{\mathcal{C}^{\boldsymbol{F}}, \boldsymbol{F}}(1^n) = 1\right] \quad . \tag{1}$$

We consider the following sequential distinguisher $\mathcal{D}_{\text{seq}}$ interacting with a pair of oracles (G, F) which can be either $(\boldsymbol{G}, \mathcal{S}_{\text{seq}}^{\boldsymbol{G}})$ or $(\mathcal{C}^{\boldsymbol{F}}, \boldsymbol{F})$. $\mathcal{D}_{\text{seq}}$ runs $\mathcal{D}_{\text{pub}}$ (see Figure 4). $\mathcal{D}_{\text{seq}}$ simply relays any F-query of $\mathcal{D}_{\text{pub}}$ to its own F oracle, returning the corresponding answer. When $\mathcal{D}_{\text{pub}}$ makes a G-query x, $\mathcal{D}_{\text{seq}}$ makes all the necessary F-queries to its own F-oracle to compute $\mathcal{C}^F(x)$ and returns this value as the answer to $\mathcal{D}_{\text{pub}}$. Once $\mathcal{D}_{\text{pub}}$ has returned 0 or 1, $\mathcal{D}_{\text{seq}}$ makes all the G-queries that have been made by $\mathcal{D}_{\text{pub}}$ to its own G-oracle and checks whether all the answers it has given to $\mathcal{D}_{\text{pub}}$ (by computing $\mathcal{C}^F$ with its own F-oracle) correspond (in which case we say that event `check` happens). If this is the case,

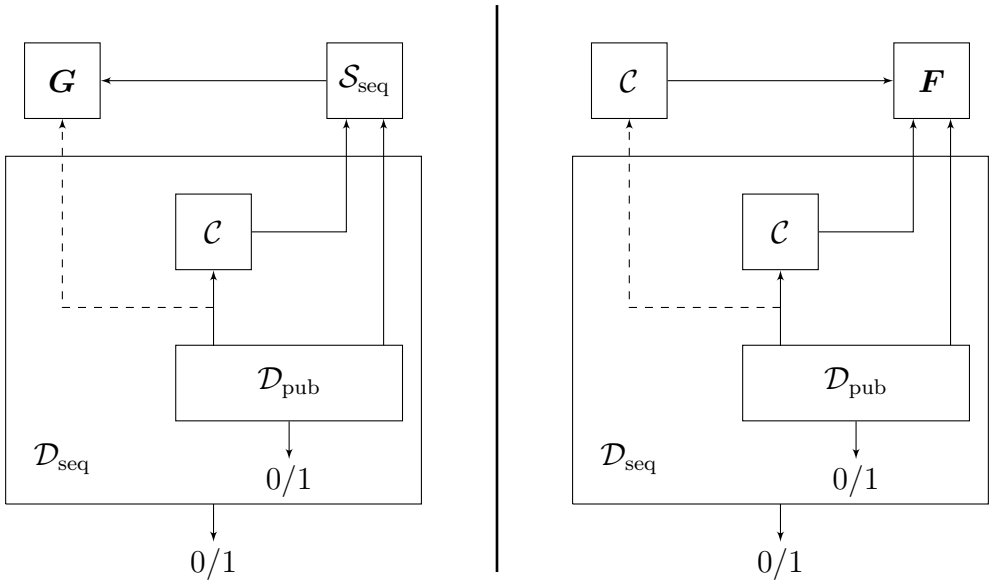

Fig. 4. Illustration of the proof of Theorem 1. The dashed arrow means that $\mathcal{D}_{\text{seq}}$ makes the corresponding queries once $\mathcal{D}_{\text{pub}}$ has returned and compares the answers with the one it computed with $\mathcal{C}$.

$\mathcal{D}_{\text{seq}}$ returns the same answer as $\mathcal{S}_{\text{pub}}$. Otherwise it returns 1. Note that $\mathcal{D}_{\text{seq}}$ is indeed sequential, and that its total oracle queries cost is less than $2q$ when $\mathcal{D}_{\text{pub}}$'s total oracle queries cost is less than q.

First, it is straightforward to verify that when $\mathcal{D}_{\text{seq}}$ interacts with $(\mathcal{C}^{\boldsymbol{F}}, \boldsymbol{F})$, `check` happens with probability one so that

$$\Pr\left[\mathcal{D}_{\text{seq}}^{\mathcal{C}^{\boldsymbol{F}},\boldsymbol{F}}(1^n)=1\right] = \Pr\left[\mathcal{D}_{\text{pub}}^{\mathcal{C}^{\boldsymbol{F}},\boldsymbol{F}}(1^n)=1\right] \ .$$

When $\mathcal{D}_{\text{seq}}$ interacts with $(\boldsymbol{G}, \mathcal{S}_{\text{seq}}^{\boldsymbol{G}})$, one can write:

$$\Pr\left[\mathcal{D}_{\text{seq}}^{\boldsymbol{G},\mathcal{S}_{\text{seq}}^{\boldsymbol{G}}}(1^n)=1\right] = \Pr\left[\mathcal{D}_{\text{seq}}^{\boldsymbol{G},\mathcal{S}_{\text{seq}}^{\boldsymbol{G}}}(1^n)=1 \,|\, \texttt{check}\right]\Pr[\texttt{check}] + \Pr[\overline{\texttt{check}}] \ .$$

Note that when event `check` occurs, all answers to F- and G-queries of $\mathcal{D}_{\text{pub}}$ have been answered as if $\mathcal{D}_{\text{pub}}$ had been interacting directly with $(\boldsymbol{G}, \mathcal{S}_{\text{pub}}^{\overline{\boldsymbol{G}}})$. This follows from the definition of $\mathcal{S}_{\text{pub}}$ and the fact that $\boldsymbol{G}$ is stateless. The statelessness of $\boldsymbol{G}$ is crucial here since even when `check` happens, the sequence of queries to $\boldsymbol{G}$ when $\mathcal{D}_{\text{seq}}$ interacts with $(\boldsymbol{G}, \mathcal{S}_{\text{seq}}^{\boldsymbol{G}})$ is not necessarily the same as when $\mathcal{D}_{\text{pub}}$ interacts with $(\boldsymbol{G}, \mathcal{S}_{\text{pub}}^{\overline{\boldsymbol{G}}})$: in the former, the G-queries of $\mathcal{D}_{\text{pub}}$ are forwarded to $\boldsymbol{G}$ by $\mathcal{D}_{\text{seq}}$ only once $\mathcal{D}_{\text{pub}}$ has returned, whereas in the later $\boldsymbol{G}$ receives $\mathcal{D}_{\text{pub}}$'s queries immediately. Hence we have:

$$\begin{aligned}
\Pr\left[\mathcal{D}_{\text{seq}}^{\boldsymbol{G},\mathcal{S}_{\text{seq}}^{\boldsymbol{G}}}(1^n)=1\right] &= \Pr\left[\mathcal{D}_{\text{pub}}^{\boldsymbol{G},\mathcal{S}_{\text{pub}}^{\overline{\boldsymbol{G}}}}(1^n)=1 \,|\, \texttt{check}\right]\Pr[\texttt{check}] + \Pr[\overline{\texttt{check}}] \\
&= \Pr\left[\mathcal{D}_{\text{pub}}^{\boldsymbol{G},\mathcal{S}_{\text{pub}}^{\overline{\boldsymbol{G}}}}(1^n)=1\right] + \Pr[\overline{\texttt{check}}]\left(1 - \Pr\left[\mathcal{D}_{\text{pub}}^{\boldsymbol{G},\mathcal{S}_{\text{pub}}^{\overline{\boldsymbol{G}}}}(1^n)=1 \,|\, \overline{\texttt{check}}\right]\right) \\
&\geq \Pr\left[\mathcal{D}_{\text{pub}}^{\boldsymbol{G},\mathcal{S}_{\text{pub}}^{\overline{\boldsymbol{G}}}}(1^n)=1\right] \ .
\end{aligned}$$

It follows from assumption (1) that

$$\left|\Pr\left[\mathcal{D}_{\text{pub}}^{\boldsymbol{G},\mathcal{S}_{\text{pub}}^{\overline{\boldsymbol{G}}}}(1^n)=1\right] - \Pr\left[\mathcal{D}_{\text{pub}}^{\mathcal{C}^{\boldsymbol{F}},\boldsymbol{F}}(1^n)=1\right]\right| \leq \left|\Pr\left[\mathcal{D}_{\text{seq}}^{\boldsymbol{G},\mathcal{S}_{\text{seq}}^{\boldsymbol{G}}}(1^n)=1\right] - \Pr\left[\mathcal{D}_{\text{seq}}^{\mathcal{C}^{\boldsymbol{F}},\boldsymbol{F}}(1^n)=1\right]\right| \ ,$$

which is less than ϵ since by hypothesis $\mathcal{C}^{\boldsymbol{F}}$ is $(2q, \sigma, \epsilon)$-seq-indifferentiable from $\boldsymbol{G}$. The result follows by noting that $\mathcal{S}_{\text{pub}}$ makes at most q `Reveal` queries and σ queries to $\boldsymbol{G}$.

Clearly, $\mathcal{S}_{\text{pub}}$ and $\mathcal{D}_{\text{pub}}$ are polynomial-time if $\mathcal{S}_{\text{seq}}$ and $\mathcal{D}_{\text{seq}}$ are, so that pub-indifferentiability holds computationally if seq-indifferentiability does. □

B Counter-Example to Theorem 1 for Stateful Ideal Primitives

When the ideal primitive $\boldsymbol{G}$ is stateful, then seq-indifferentiability does not necessarily imply pub-indifferentiability in the computational setting, as was observed by Ristenpart. To see this, let $\mathcal{E} = (\texttt{KeyGen}, \texttt{Enc}, \texttt{Dec})$ be a IND-CPA public-key encryption scheme. The ideal primitive $\boldsymbol{G}$ maintains a hashtable T with n-bit keys and takes as input an n-bit string x and a public key $\texttt{pk}$ for $\mathcal{E}$.

procedure $\boldsymbol{G}(x,\texttt{pk})$
  if $T(x) = \perp$ **then**
    $y \leftarrow_{\mathcal{R}} \{0,1\}^n$
    $T(x) := \texttt{Enc}_{\texttt{pk}}(y)$
  end if
  return $T(x)$
end procedure

The construction $\mathcal{C}$ is quite similar to $\boldsymbol{G}$, but instead of drawing a uniformly random y it uses a random function oracle $\boldsymbol{F} : \{0,1\}^n \to \{0,1\}^n$:

procedure $\mathcal{C}^{\boldsymbol{F}}(x,\texttt{pk})$
  if $T(x) = \perp$ **then**
    $y := \boldsymbol{F}(x)$
    $T(x) := \texttt{Enc}_{\texttt{pk}}(y)$
  end if
  return $T(x)$
end procedure

One can show that $\mathcal{C}^{\boldsymbol{F}}$ is strongly, computationally seq-indifferentiable from $\boldsymbol{G}$, but not pub-indifferentiable. The idea is that in the seq-indifferentiability game, the simulator can always get the y values drawn by $\boldsymbol{G}$ by generating the public keys by itself, whereas in the pub-indifferentiability game, when the distinguisher makes a G-query before the simulator, the y value will be hidden to $\mathcal{S}_{\text{pub}}$ unless it can break the one-wayness of $\mathcal{E}$. The seq-indifferentiability simulator maintains an history F for the simulated oracle and is defined as follows:

procedure $\mathcal{S}^{\boldsymbol{G}}_{\text{seq}}(x)$
  if $F(x) = \perp$ **then**
    $(\texttt{pk}, \texttt{sk}) \leftarrow \texttt{KeyGen}(1^n)$
    $c := \boldsymbol{G}(x, \texttt{pk})$
    $y := \texttt{Dec}_{\texttt{sk}}(c)$
    $F(x) := y$
  end if
  return $F(x)$
end procedure

It is not very hard to see that the above simulator works for seq-indifferentiability. On the other hand, consider the following distinguisher for pub-indifferentiability:

procedure $\mathcal{D}^{\Theta_1,\Theta_2}_{\text{pub}}(1^n)$
  $(\texttt{pk}, \texttt{sk}) \leftarrow \texttt{KeyGen}(1^n)$
  $c := \Theta_1(0, \texttt{pk})$
  $y := \texttt{Dec}_{\texttt{sk}}(c)$
  $y' := \Theta_2(0)$
  if $y = y'$ **then**

```
        return 1
    else
        return 0
    end if
end procedure
```

When $(\Theta_1, \Theta_2) = (\mathcal{C}^{\boldsymbol{F}}, \boldsymbol{F})$ then $\mathcal{D}_{\text{pub}}$ always returns 1. However any efficient simulator $\mathcal{S}_{\text{pub}}$ such that $\mathcal{D}_{\text{pub}}$ returns 1 with non negligible probability when $(\Theta_1, \Theta_2) = (\boldsymbol{G}, \mathcal{S}_{\text{pub}}^{\boldsymbol{G}})$ can be turned into an algorithm breaking the one-wayness of $\mathcal{E}$. Hence $\mathcal{D}_{\text{pub}}$ distinguishes the two systems with overwhelming probability.

C Sequential Distinguisher for the 5-Round Feistel Construction

The sequential distinguisher $\mathcal{D}$ proceeds as follows (see Figure 5). It chooses an arbitrary value Z_{13}, two arbitrary values Y_{14} et Y_{23}, and queries $F_3(Y_{14})$ and $F_3(Y_{23})$. It then computes:

$$\begin{cases} X_{12} = Z_{13} \oplus F_3(Y_{14}) \\ X_{34} = Z_{13} \oplus F_3(Y_{23}) \end{cases}.$$

Notations are chosen such that input round values sharing a common index correspond to the same input-output pair of the Feistel scheme: we say they constitute a chain. For example, (X_{12}, Y_{14}, Z_{13}) constitute a chain since $X_{12} = Z_{13} \oplus F_3(Y_{14})$.
The distinguisher then queries $F_2(X_{12})$ and $F_2(X_{34})$ and computes:

$$\begin{cases} R_1 = Y_{14} \oplus F_2(X_{12}) \\ R_2 = Y_{23} \oplus F_2(X_{12}) \\ R_3 = Y_{23} \oplus F_2(X_{34}) \\ R_4 = Y_{14} \oplus F_2(X_{34}) \end{cases}.$$

Note that necessarily $R_1 \oplus R_2 \oplus R_3 \oplus R_4 = 0$.
Then the distinguisher queries $F_1(R_1)$, $F_1(R_2)$, $F_1(R_3)$, and $F_1(R_4)$ and computes:

$$\begin{cases} L_1 = X_{12} \oplus F_1(R_1) \\ L_2 = X_{12} \oplus F_1(R_2) \\ L_3 = X_{34} \oplus F_1(R_3) \\ L_4 = X_{34} \oplus F_1(R_4) \end{cases}.$$

Finally the distinguisher makes the P-queries $(S_1, T_1) = P(0, (L_1, R_1))$, $(S_2, T_2) = P(0(L_2, R_2))$, $(S_3, T_3) = P(0, (L_3, R_3))$ and $(S_4, T_4) = P(0, (L_4, R_4))$. If $S_1 \oplus S_2 \oplus S_3 \oplus S_4 = 0$, it returns 1, otherwise it returns 0. Note that this distinguisher is sequential.

First, one can easily verify that $\mathcal{D}$ always returns 1 when it interacts with $(\Psi_5^{\boldsymbol{F}}, \boldsymbol{F})$. Indeed, denote $Z_{24} = X_{12} \oplus F_3(Y_{23})$ the input value to F_4 associated with (L_2, R_2). Since $X_{12} \oplus F_3(Y_{14}) = X_{34} \oplus F_3(Y_{23}) = Z_{13}$, then $Z_{24} = X_{34} \oplus F_3(Y_{14})$, so that Z_{24} is also the input value to F_4 associated with (L_4, R_4). It follows that:

$$\begin{cases} S_1 = Y_{14} \oplus F_4(Z_{13}) \\ S_2 = Y_{23} \oplus F_4(Z_{24}) \\ S_3 = Y_{23} \oplus F_4(Z_{13}) \\ S_4 = Y_{14} \oplus F_4(Z_{24}) \end{cases},$$

and the relation $S_1 \oplus S_2 \oplus S_3 \oplus S_4 = 0$ is always verified.

On the contrary, when interacting with $(\boldsymbol{P}, \mathcal{S}^{\boldsymbol{P}})$, it returns 1 only with negligible probability. Indeed, considering the union of $\mathcal{D}$ and $\mathcal{S}$ as a single machine making a polynomial number of queries to the random permutation $\boldsymbol{P}$, it can find four input/output pairs $(S_i, T_i) = \boldsymbol{P}(0, (L_i, R_i))$ satisfying $R_1 \oplus R_2 \oplus R_3 \oplus R_4 = 0$ and $S_1 \oplus S_2 \oplus S_3 \oplus S_4 = 0$ only with negligible probability.

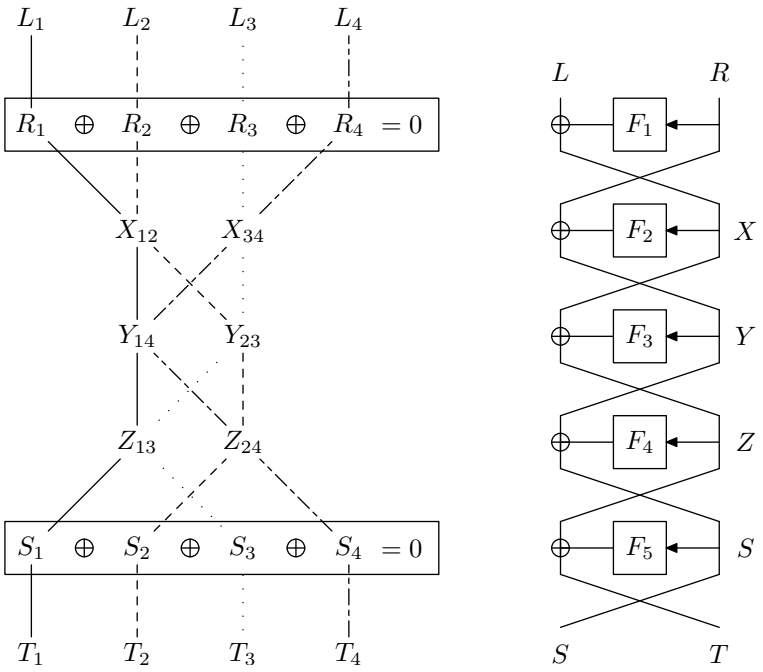

Fig. 5. Description of the sequential distinguisher for the 5-round Feistel construction.

D Separating Correlation Intractability and Sequential Indifferentiability

According to Theorem 3, sequential indifferentiability implies correlation intractability. However, it does not hold the other way around. Below we give a constructive counter-example.

Let $\mathcal{C}^{\boldsymbol{E}}$ be a construction based on some ideal primitive $\boldsymbol{E}$ which is seq-indifferentiable from a random function $\boldsymbol{F} = (\boldsymbol{F}_n)$, $\boldsymbol{F}_n : \{0,1\}^n \to \{0,1\}^n$. By Theorem 3, $\mathcal{C}^{\boldsymbol{E}}$ is also correlation intractable with respect to the random function $\boldsymbol{F}$.

Now consider the primitive $\boldsymbol{G} = (\boldsymbol{G}_n)$ where $\boldsymbol{G}_n : \{0,1\}^n \to \{0,1\}^n$ is such that $\boldsymbol{G}_n(0^n) = 0^n$ and for $x \in \{0,1\}^n \setminus \{0^n\}$, we have $\boldsymbol{G}_n(x) = \boldsymbol{F}_n(x)$. Let $\mathcal{R}$ be be a relation evasive with respect to the ideal primitive $\boldsymbol{G}$. Clearly, $\mathcal{R}$ is also evasive with respect to the random function $\boldsymbol{F}$ so that $\mathcal{C}^{\boldsymbol{E}}$ is also correlation intractable with respect to $\boldsymbol{G}$. However, $\mathcal{C}^{\boldsymbol{E}}$ is not seq-indifferentiable from $\boldsymbol{G}$. Indeed, consider a distinguisher which simply makes the query 0^n to its left oracle: the answer will be 0^n when it is $\boldsymbol{G}$, and will be 0^n with only negligible probability when it is $\mathcal{C}^{\boldsymbol{E}}$ (since otherwise this would yield a sequential distinguisher distinguishing $\mathcal{C}^{\boldsymbol{E}}$ from $\boldsymbol{F}$).

E Implications for Chosen-Key and Known-Key Attacks on Block Ciphers

Knudsen and Rijmen [KR07] have introduced the model of known-key attacks on block ciphers, where the attacker is given a random or chosen key K to the block cipher, and must find inputs to the block cipher that together with their image satisfy a relation that would be hard to satisfy for a random invertible permutation. In other words, the attacker must break the correlation intractability of the block cipher for that particular key.

In the random oracle model, there are at least two straightforward ways to obtain a block cipher with a Feistel construction. The first one is to let round functions have input length $\ell+n$, where ℓ is the key length, and to prepend the key K to the input of each round function. Another way is to xor keys $(k_1, \ldots, k_r)$ (where $|k_i| = n$) to the input of the r round functions $(\boldsymbol{F}_1, \ldots, \boldsymbol{F}_r)$ (the pseudorandomness

of this construction in the random oracle model has for example been studied by [GR04]). Many variations can be explored, *e.g.* having a single round function $\boldsymbol{F}$ instead of independent ones, having $\boldsymbol{F}_i$'s be random invertible permutations rather that random functions, etc.

An interesting result of [KR07] is that for a 7-round Feistel construction using a single random invertible permutation $\boldsymbol{P}$ as round function, and independent keys $(k_1, \ldots, k_7)$ xored to the input of $\boldsymbol{P}$ at each round, then the resulting block cipher is not correlation intractable (even when the keys are only random and known from the attacker, not chosen): namely with high probability on the choice of the keys, the attacker can find inputs (L, R) and (L', R') that together with the corresponding outputs (S, T) and (S', T') satisfy $R \oplus R' \oplus T \oplus T' = 0$.

Our results on the correlation intractability of the 6-round Feistel construction shows that the block cipher obtained by prepending the key to the input of each round function is correlation intractable (in the random oracle model), and hence immune to known-key and even chosen-key attacks. Other variations need more careful analysis. In particular, note that the variant using a single round function is clearly not immune to known or chosen key attacks (at least when the same key is used at each round): for example for any number of rounds, $\Psi_r^{\boldsymbol{F},\ldots,\boldsymbol{F}}(L, R) = (S, T)$ implies $\Psi_r^{\boldsymbol{F},\ldots,\boldsymbol{F}}(T, S) = (R, L)$ (this is in fact true for any palindromic sequence of round functions).

F Seq-Indifferentiability Beyond the Birthday Barrier for the Construction of [CDMS10]

Coron *et al.* [CDMS10] considered the 3-round permutation $\Psi_3 : \{0,1\}^{2n} \to \{0,1\}^{2n}$ defined as follows (see Figure 6 for an illustration), given block ciphers E_1, E_2 and E_3 with n-bit key (first variable) and n-bit input/output (second variable):

$$\begin{aligned} X &= E_1(R, L) \\ S &= E_2(X, R) \\ T &= E_3(S, X) \\ \Psi_3(L, R) &:= (S, T) \end{aligned}$$

The 3 round block cipher $\Psi_3' : \{0,1\}^k \times \{0,1\}^{2n} \to \{0,1\}^{2n}$ is defined as follows, given block ciphers E_1, E_2 and E_3 with $(k+n)$-bit key and n-bit input/output:

$$\begin{aligned} X &= E_1(K\|R, L) \\ S &= E_2(K\|X, R) \\ T &= E_3(K\|S, X) \\ \Psi_3'(K, (L, R)) &:= (S, T) \end{aligned}$$

We now state our main result in this section: the 3-round Feistel construction $(\{0,1\}^{2n} \to \{0,1\}^{2n})$ is seq-indifferentiable from a random permutation up to $q \ll 2^n$ queries. To get an ideal cipher, it suffices to prepend a key K to the 3 ideal ciphers E_1, E_2 and E_3; one then gets a family of independent random permutation, parametrized by K, *i.e.* an ideal cipher.

Theorem 4. *The 3-round Feistel construction Ψ_3 is (q, σ, ε)-seq-indifferentiable from a random invertible permutation $P : \{0,1\}^{2n} \to \{0,1\}^{2n}$, with $\sigma(q) = nq$, and $\varepsilon = \mathcal{O}(q/2^n)$. The running time of the simulator for answering a single query is $t_S = \mathcal{O}(n \log q)$.*

We only consider the 3-round permutation Ψ_3. The extension to block-cipher Ψ_3' is straightforward. We must construct a simulator $\mathcal{S}$ such that the two systems formed by (Ψ_3, E) and $(P, \mathcal{S})$ are indistinguishable.

Our simulator maintains a history of already answered queries for E_1, E_2 and E_3. Formally, $(1, R, L, X)$ exists in history if and only if the simulator has answered $E_1(R, L)$ query as X or $E_1^{-1}(R, X)$ query as L previously. Similar conditions hold for $(2, X, R, S)$ and $(3, S, X, T)$ as well. We define the following algorithms:

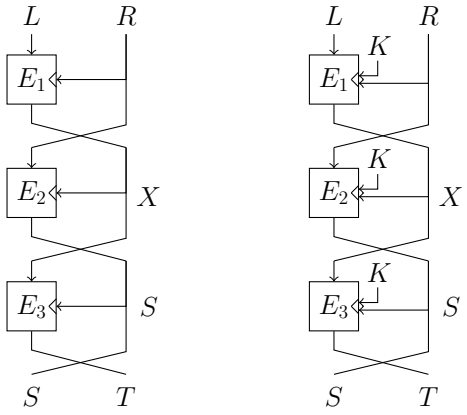

Fig. 6. 3-round permutation $\Psi_3(L,R)$ (left) and 3-round block-cipher $\Psi'_3(K,(L,R))$ (right)

- $\mathsf{GuessE_1}(R,L)$ returns a random $X \in \{0,1\}^n \setminus \mathcal{B}$ where $\mathcal{B}$ is the set of already defined values for $E_1(R,\cdot)$. Algorithms $\mathsf{GuessE_1^{-1}}$, $\mathsf{GuessE_3}$ and $\mathsf{GuessE_3^{-1}}$ work in a similar fashion.
- $\mathsf{ConflictE_1}(R,L,X)$ returns TRUE iff $(1,R,L,*)$ or $(1,R,*,X)$ exist in history. In other words, it checks whether $E_1(R,L)$ or $E_1^{-1}(R,X)$ has been defined before. $\mathsf{ConflictE_2}$ and $\mathsf{ConflictE_3}$ work in a similar fashion.
- $\mathsf{Store}(L,R,X,S,T)$ saves $(1,R,L,X)$, $(2,X,R,S)$ and $(3,S,X,T)$ in history.

The distinguisher's queries are answered as follows by the simulator:

$E_1(R,L)$ query:
1. IF $(1,R,L,*) \in$ history
2. RETURN $*$
3. $X \leftarrow \mathsf{GuessE_1}(R,L)$
4. $(\mathsf{Err},S,T) \leftarrow \mathsf{Check}(L,R,X)$
5. IF Err = TRUE GOTO 3
6. $\mathsf{Store}(L,R,X,S,T)$
7. RETURN X

$E_1^{-1}(R,X)$ query
1. IF $(1,R,*,X) \in$ history
2. RETURN $*$
3. $L \leftarrow \mathsf{GuessE_1^{-1}}(R,X)$
4. $(\mathsf{Err},S,T) \leftarrow \mathsf{Check}(L,R,X)$
5. IF Err = TRUE GOTO 3
6. $\mathsf{Store}(L,R,X,S,T)$
7. RETURN L

$E_2(X,R)$ query:
1. IF $(2,X,R,*) \in$ history
2. RETURN $*$
3. $L \leftarrow \mathsf{GuessE_1^{-1}}(R,X)$
4. $(\mathsf{Err},S,T) \leftarrow \mathsf{Check}(L,R,X)$
5. IF Err = TRUE GOTO 3
6. $\mathsf{Store}(L,R,X,S,T)$
7. RETURN S

$\mathsf{Check}(L,R,X)$:
1. $S\|T \leftarrow P(L\|R)$
2. IF $\mathsf{ConflictE_2}(X,R,S)$ = TRUE OR $\mathsf{ConflictE_3}(S,X,T)$ = TRUE
3. RETURN (TRUE, $*$, $*$)
4. RETURN (FALSE, S, T).

The procedure for answering the other queries is essentially symmetric; we provide it for completeness:

$E_3^{-1}(S,T)$ query:
1. IF $(3,S,*,T) \in$ history
2. RETURN $*$
3. $X \leftarrow \mathsf{GuessE_3^{-1}}(S,T)$
4. $(\mathsf{Err},L,R) \leftarrow \mathsf{Check}^{-1}(S,T,X)$
5. IF Err = TRUE GOTO 3
6. $\mathsf{Store}(L,R,X,S,T)$
7. RETURN X

$E_3(S,X)$ query
1. IF $(3,S,X,*) \in$ history
2. RETURN $*$
3. $T \leftarrow \mathsf{GuessE_3}(S,X)$
4. $(\mathsf{Err},L,R) \leftarrow \mathsf{Check}^{-1}(S,T,X)$
5. IF Err = TRUE GOTO 3
6. $\mathsf{Store}(L,R,X,S,T)$
7. RETURN T

$E_2^{-1}(X,S)$ query:
1. IF $(2,X,*,S) \in$ history
2. RETURN $*$
3. $T \leftarrow \mathsf{GuessE_3}(S,X)$
4. $(\mathsf{Err},L,R) \leftarrow \mathsf{Check}^{-1}(S,T,X)$
5. IF Err = TRUE GOTO 3
6. $\mathsf{Store}(L,R,X,S,T)$
7. RETURN R

$\mathsf{Check}^{-1}(S,T,X)$:
1. $L\|R \leftarrow P^{-1}(S\|T)$
2. If $\mathsf{ConflictE}_2(X,R,S) = \textsc{True}$ Or $\mathsf{ConflictE}_1(R,L,X) = \textsc{True}$
3. Return $(\textsc{True},*,*)$
4. Return $(\textsc{False},L,R)$.

For a given set of queries Q and their responses $\mathcal{X}(Q)$ we define the *extender consistency* as the property that the responses to Ψ_3 (or P) are equal to those that one would obtain by applying the extender construction from responses to E (or $\mathcal{S}$) (when queries to E or $\mathcal{S}$ suffice to preform the calculation). By construction, the system (Ψ_3, E) always gives consistent response. Also from the definition of our simulator $\mathcal{S}$, it is evident that $(P,\mathcal{S})$ gives consistent responses as well ($\mathcal{S}$ runs in polynomial time unless it enters an exponential loop with a small probability). For any distinguisher $\mathcal{D}$ against the systems (Ψ_3, E) and $(P,\mathcal{S})$ we construct another distinguisher $\mathcal{D}''$ by the following way.

1. $\Psi_3(L,R) \to (S,T)$ query in $\mathcal{D}$ is replaced in $\mathcal{D}''$ by the sequence of queries
$$E_1(R,L) \to X, E_2(X,R) \to S, E_3(S,X) \to T.$$
2. $\Psi_3^{-1}(S,T) \to (L,R)$ query in $\mathcal{D}$ is replaced in $\mathcal{D}''$ by the sequence of queries
$$E_3^{-1}(S,T) \to X, E_2^{-1}(X,S) \to R, E_1^{-1}(R,X) \to L.$$

We argue that $\mathcal{D}''$ is more powerful than $\mathcal{D}$. In replacement 1, $\mathcal{D}''$ actually observes the 5-tuple (L,R,X,S,T), whereas $\mathcal{D}$ can only observe the 4-tuple (L,R,S,T). Moreover, the systems (Ψ_3,E) and $(P,\mathcal{S})$ always give extender-consistent responses. Hence, $\mathcal{D}''$ observes the exact same information as observed by $\mathcal{D}$, plus some extra information. Formally, we have the following theorem,

Theorem 5. *For any distinguisher $\mathcal{D}$ and the distinguisher $\mathcal{D}''$ constructed in the above way, we have*
$$\mathsf{Adv}^{\mathcal{D}} \le \mathsf{Adv}^{\mathcal{D}''}.$$
Moreover, $\mathcal{D}''$ only makes queries to E or $\mathcal{S}$.

Remark 4. The above theorem is not true in case of general indifferentiability, this kind of query replacement would mean the simulator is getting some extra information, namely the queries to the P oracle.

For any distinguisher $\mathcal{D}''$ making only E or $\mathcal{S}$ queries we construct another distinguisher D' by the following way.

1. $E_1(R,L) \to X$ query in $\mathcal{D}''$ is replaced by the sequence of queries $E_1(R,L) \to X$, $E_2(X,R) \to S$, $E_3(S,X) \to T$ in $\mathcal{D}'$.
2. $E_1^{-1}(R,X) \to L$ query in $\mathcal{D}''$ is replaced by the sequence of queries $E_1^{-1}(R,X) \to L$, $E_2(X,R) \to S$, $E_3(S,X) \to T$ in $\mathcal{D}'$.
3. $E_2(X,R) \to L$ query in $\mathcal{D}''$ is replaced by the sequence of queries $E_1^{-1}(R,X) \to L$, $E_2(X,R) \to S$, $E_3(S,X) \to T$ in $\mathcal{D}'$.
4. The queries $E_3^{-1}(S,T)$, $E_3(S,X)$ and $E_2^{-1}(X,S)$ are processed essentially symmetrically.
5. $\mathcal{D}'$ does not make any duplicate or *trivial* query. (Queries which try to verify whether E_i is a well defined permutation are trivial queries).

In general, $\mathcal{D}'$ always observes the exact same or more information than $\mathcal{D}''$. It might happen that for some (L,R,S,T) such that $\Psi_3(L,R) = (S,T)$, $\mathcal{D}'$ gets the intermediate X value through $E_1(R,L)$ query, whereas $\mathcal{D}''$ finds it through $E_3^{-1}(S,T)$ query or vice-versa. When the distinguishers are interacting with (Ψ_3,E) the X value is always the same irrespective of whether it is retrieved through E_1 or E_3^{-1} query. And when the distinguishers are interacting with $(P,\mathcal{S})$, X follows the exact same probability distribution in both scenarios. This is due to the fact that, for fixed (L,R,S,T) while answering $E_1(R,L)$ or $E_3^{-1}(S,T)$ the simulator picks X uniformly over all possible X which do not conflict with previous responses. Proof of Lemma 12 explains it in more details. Hence we have the following theorem.

Theorem 6. *For any distinguisher $\mathcal{D}$ and the distinguishers $\mathcal{D}''$ and $\mathcal{D}'$ (making only E or $\mathcal{S}$ queries) constructed in the above way, we have*

$$\mathsf{Adv}^{\mathcal{D}} \leq \mathsf{Adv}^{\mathcal{D}''} \leq \mathsf{Adv}^{\mathcal{D}'}.$$

Moreover, if $\mathcal{D}$ makes at most q queries to the systems (Ψ_3, E) or $(P, \mathcal{S})$, then $\mathcal{D}'$ only makes at most q many 3-query sequences to (Ψ_3, E) or $(P, \mathcal{S})$. Each query sequence is one of the following types.

- Type I $\quad E_1(R, L) \to X,\ E_2(X, R) \to S,\ E_3(S, X) \to T$
- Type II $\quad E_1^{-1}(R, X) \to L,\ E_2(X, R) \to S,\ E_3(S, X) \to T$
- Type III $\quad E_3^{-1}(S, T) \to X,\ E_2^{-1}(X, S) \to R,\ E_1^{-1}(R, X) \to L$
- Type IV $\quad E_3(S, X) \to T,\ E_2^{-1}(X, S) \to R,\ E_1^{-1}(R, X) \to L$

Type I and Type III query sequences are actually symmetric. The same is true for Type II and IV query sequences as well. We write $\{0,1\}^n$ as $\mathbf{Y}$. If $\mathsf{Adv}_{i+1}^{\mathcal{D}'}$ is the advantage of the distinguisher $\mathcal{D}'$ for the $(i+1)^{th}$ 3-query sequence we have the following two theorems.

Theorem 7. *If the $(i+1)^{th}$ 3-query sequence made by $\mathcal{D}'$ is either of* Type I *or* Type III *we have,*

$$\mathsf{Adv}_{i+1}^{\mathcal{D}'} \leq \frac{2i}{|\mathbf{Y}|^2}.$$

Theorem 8. *If the $(i+1)^{th}$ 3-query sequence made by $\mathcal{D}'$ is either of* Type II *or* Type IV *we have,*

$$\mathsf{Adv}_{i+1}^{\mathcal{D}'} \leq \frac{5i}{|\mathbf{Y}|^2} + \frac{25i^2}{|\mathbf{Y}|^3} + \frac{4i^3}{|\mathbf{Y}|^4}.$$

We know, $\mathsf{Adv}^{\mathcal{D}'} \leq \sum_{i=0}^{q-1} \mathsf{Adv}_{i+1}^{\mathcal{D}'}$. Hence Theorem 6, Theorem 7 and Theorem 8 together complete the proof of Theorem 4.

F.1 Proof of Theorem 7 and Theorem 8

Input-ouput of each 3-query sequence made by $\mathcal{D}'$ is actually a 5-tuple (L, R, X, S, T). In fact, input of each 3-query sequence is a 2-tuple and output is a 3-tuple. Say before $(i+1)^{th}$ query, $\mathcal{D}'$ has observed i many such distinct (as $\mathcal{D}'$ does not make duplicate or trivial queries) 5-tuples $(L_j, R_j, X_j, S_j, T_j)$ for $j = 1$ to i. When $\mathcal{D}'$ is interacting with $(P, \mathcal{S})$, the simulator's internal history also contains exactly the same information. Let $\mathbf{L}$, $\mathbf{R}$, $\mathbf{X}$, $\mathbf{S}$ and $\mathbf{T}$ be the set of L_j's, R_j's, X_j's, S_j's and T_j's (for $j = 1$ to i) respectively. We partition the set $\mathbf{L}$ as

$$\mathbf{L} = \mathbf{L}^1 \cup \mathbf{L}^2 \cup \cdots \cup \mathbf{L}^i,$$

such that $\ell \in \mathbf{L}^k$ if and only if ℓ has appeared exactly k times in history (or there are exactly k-many j values such that $\ell = L_j$). We do similar partitioning for the sets $\mathbf{R}$, $\mathbf{X}$, $\mathbf{S}$ and $\mathbf{T}$ as well. Note,

$$\sum_j j|\mathbf{L}^j| = \sum_j j|\mathbf{R}^j| = \sum_j j|\mathbf{X}^j| = \sum_j j|\mathbf{S}^j| = \sum_j j|\mathbf{T}^j| = i. \tag{2}$$

We also define $\mathbf{L}^0 = \mathbf{Y} \setminus \mathbf{L}$. $\mathbf{R}^0$, $\mathbf{X}^0$, $\mathbf{S}^0$ and $\mathbf{T}^0$ are defined similarly. The proofs of Theorem 7 and Theorem 8 are essentially independent. We describe Theorem 8 before, because the proof is simpler.

Theorem 8

We prove the result when the query sequence is of TYPE II. For TYPE IV query sequence the result follows because of symmetry. Let (R, X) be the input to the TYPE II query sequence. Note, we can not have $(R, X) = (R_j, X_j)$ for some $j \in [1, i]$, because then $E_1^{-1}(R, X)$ becomes duplicate or trivial query. (L^E, S^E, T^E) and (L^P, S^P, T^P) be the random variables corresponding to the output tuple depending on whether $\mathcal{D}'$ is interacting with (Ψ_3, E) or $(P, \mathcal{S})$ respectively.

Let $\mathbf{B}_L$ be the set of values for which $E_1(R, .)$ is defined in history. Also on $\mathbf{B}_S$, $E_2^{-1}(X, .)$ is defined. If $R \in \mathbf{R}^{i_1}$ and $X \in \mathbf{X}^{i_2}$, we have

$$|\mathbf{B}_L| = i_1 \text{ and } |\mathbf{B}_S| = i_2,$$
$$\mathbf{B}_L \subseteq \mathbf{L} \text{ and } \mathbf{B}_S \subseteq \mathbf{S}.$$

$\mathbf{OP}_{ST}$ be the set of (S_j, T_j) tuples present in history. Also for $j = 1, \cdots, i$, (S_j, T_j)'s are actually distinct. In other words, $|\mathbf{OP}_{ST}| = i$.

Lemma 9. *If $R \in \mathbf{R}^{i_1}$ and $X \in \mathbf{X}^{i_2}$ then for $j = 0, \cdots, i$ we have*

$$\Pr[(L^E, S^E, T^E) = (L, S, T)] = \frac{1}{|\mathbf{Y}| - i_1} \times \frac{1}{|\mathbf{Y}| - i_2} \times \frac{1}{|\mathbf{Y}| - j}$$

when $(L, S, T) \in (\mathbf{Y} \setminus \mathbf{B}_L) \times ((\mathbf{S}^j \times \mathbf{Y}) \setminus \mathbf{OP}_{ST})$. Moreover,

$$|(\mathbf{Y} \setminus \mathbf{B}_L) \times (((\mathbf{S}^j \setminus \mathbf{B}_S) \times \mathbf{Y}) \setminus \mathbf{OP}_{ST})| = (|\mathbf{Y}| - i_1)|\mathbf{S}^j \setminus \mathbf{B}_S|(|\mathbf{Y}| - j).$$

For the tuples (L, S, T) not covered by Lemma 9, $\Pr[(L^E, S^E, T^E) = (L, S, T)]$ is actually zero. In fact, (L^E, S^E, T^E) always have some non-zero probability over the set $(\mathbf{Y} \setminus \mathbf{B}_L) \times \big(\big((\mathbf{Y} \setminus \mathbf{B}_S) \times \mathbf{Y}\big) \setminus \mathbf{OP}_{ST}\big)$, even though non-uniform. We will see, (L^P, S^P, T^P) is actually uniform over the same set, and at other points it has zero probability as well. Using some basic counting principles, we also get

$$\begin{aligned}|\big((\mathbf{Y} \setminus \mathbf{B}_S) \times \mathbf{Y}\big) \setminus \mathbf{OP}_{ST}| &= (|\mathbf{Y}| - |\mathbf{B}_S|)|\mathbf{Y}| - |\mathbf{OP}_{ST}| + \sum_{j'} j'|\mathbf{B}_S \cap \mathbf{S}^{j'}| \\ &= |\mathbf{Y}|^2 - i_2|\mathbf{Y}| - i + \sum_{j'} j'|\mathbf{B}_S \cap \mathbf{S}^{j'}|.\end{aligned}$$

Formally, we have the following lemma.

Lemma 10. *If $R \in \mathbf{R}^{i_1}$ and $X \in \mathbf{X}^{i_2}$ then, (L^P, S^P, T^P) is uniform over $(\mathbf{Y} \setminus \mathbf{B}_L) \times \big(\big((\mathbf{Y} \setminus \mathbf{B}_S) \times \mathbf{Y}\big) \setminus \mathbf{OP}_{ST}\big)$. More specifically, for $(L, S, T) \in (\mathbf{Y} \setminus \mathbf{B}_L) \times \big(\big((\mathbf{Y} \setminus \mathbf{B}_S) \times \mathbf{Y}\big) \setminus \mathbf{OP}_{ST}\big)$ we have,*

$$\Pr[(L^P, S^P, T^P) = (L, S, T)] = \frac{1}{|\mathbf{Y}| - i_1} \times \frac{1}{|\mathbf{Y}|^2 - i_2|\mathbf{Y}| - i + \sum_{j'} j'|\mathbf{B}_S \cap \mathbf{S}^{j'}|}.$$

Proof. For a TYPE II query sequence the simulator response is decided by simulators behavior on $E_1^{-1}(R, X)$ query. Inside the $\mathsf{Check}(L, R, X)$ function, L is not passed to $\mathsf{ConflictE}_2$ and $\mathsf{ConflictE}_3$ functions. L, is only used as input to P. P being a random permutation the probability distributions of L^P and (S^P, T^P) are actually independent. Again, due to uniform randomness of P, (S^P, T^P) is actually uniform over a set of possible values which does conflict with history. Again, the probability that $\mathsf{Check}(L, R, X)$ returns Err is the same for all possible outputs of GuessE_1^{-1}. Also GuessE_1^{-1} outputs uniformly. Hence, the distribution of L^E is also uniform. The result follows, because joint probability distribution of two uniform and independent distribution is also uniform. □

$\mathsf{Adv}_{i+1}^{\mathcal{D}'}$ is nothing but sum of half of the probability differences of (L^E, S^E, T^E) and (L^P, S^P, T^P) at all points. Instead of considering all the points we can only consider the points where the probability

corresponding to (L^P, S^P, T^P) is bigger. Now let us calculate the probability differences. For $j = 0, \cdots, i$ if $(L, S, T) \in (\mathbf{Y} \setminus \mathbf{B}_L) \times ((\mathbf{S}^j \times \mathbf{Y}) \setminus \mathbf{OP}_{ST})$ we have,

$$\begin{aligned}
&\Pr[(L^E, S^E, T^E) = (L, S, T)] - \Pr[(L^P, S^P, T^P) = (L, S, T)] \\
&= \frac{1}{|\mathbf{Y}| - i_1} \times \frac{1}{|\mathbf{Y}| - i_2} \times \frac{1}{|\mathbf{Y}| - j} - \frac{1}{|\mathbf{Y}| - i_1} \times \frac{1}{|\mathbf{Y}|^2 - i_2|\mathbf{Y}| - i + \sum_{j'} j'|\mathbf{B}_S \cap \mathbf{S}^{j'}|} \\
&= \frac{j(|\mathbf{Y}| - i_2) - i + \sum_{j'} j'|\mathbf{B}_S \cap \mathbf{S}^{j'}|}{(|\mathbf{Y}| - i_1)(|\mathbf{Y}| - i_2)(|\mathbf{Y}| - j)(|\mathbf{Y}|^2 - i_2|\mathbf{Y}| - i + \sum_{j'} j'|\mathbf{B}_S \cap \mathbf{S}^{j'}|)}
\end{aligned}$$

- As, $i \leq |\mathbf{Y}|/2$ and $i_1 \leq i$ the expression above is bigger than zero for $j \geq 1$.
- Also, $\sum_{j'} j'|\mathbf{B}_S \cap \mathbf{S}^{j'}| \leq \sum_{j'} j'|\mathbf{S}^{j'}| \leq i$. Hence the expression above is negative for $j = 0$.

So, we can only consider $j = 0$, for calculating $\mathsf{Adv}^{\mathcal{D}'}_{i+1}$.

$$\begin{aligned}
\mathsf{Adv}^{\mathcal{D}'}_{i+1} &\leq \frac{(|\mathbf{Y}| - i_1)(|\mathbf{Y}| - |\mathbf{S}|)|\mathbf{Y}| \times i}{(|\mathbf{Y}| - i_1)(|\mathbf{Y}| - i_2)|\mathbf{Y}|(|\mathbf{Y}|^2 - i_2|\mathbf{Y}| - i + \sum_{j'} j'|\mathbf{B}_S \cap \mathbf{S}^{j'}|)} \\
&\leq \frac{i}{|\mathbf{Y}|^2 - i|\mathbf{Y}| - i} \qquad (As,\ i_2 = |\mathbf{B}_S| \leq |\mathbf{S}| \text{ and } i_2 \leq i) \\
&\leq \frac{2i}{|\mathbf{Y}|^2} \qquad (As,\ i \leq |\mathbf{Y}|/2 - 1)
\end{aligned}$$

Theorem 7

We prove the result when the query sequence is of Type I. For Type III query sequence the result follows because of symmetry. Let (L, R) be the input to the Type I query sequence. Note, we can not have $(L, R) = (L_j, R_j)$ for some $j \in [1, i]$, because then $E_1(R, L)$ becomes duplicate or trivial query. (X^E, S^E, T^E) and (X^P, S^P, T^P) be the random variables corresponding to the output tuple depending on whether $\mathcal{D}'$ is interacting with (Ψ_3, E) or $(P, \mathcal{S})$ respectively.

As before, $\mathbf{B}_X$ is the set of values for which $E_1^{-1}(R, .)$ is already defined in history. Let us assume $R \in \mathbf{R}^{i_1}$. We have

$$|\mathbf{B}_X| = i_1 \text{ and } \mathbf{B}_X \subseteq \mathbf{X}.$$

We also partition the sets $\mathbf{S}^j$'s as follows. For $j = 0, \cdots, i$,

$$\mathbf{S}^j = \mathbf{S}^j_0 \cup \mathbf{S}^j_1 \cup \cdots \cup \mathbf{S}^j_{\min(i_1, j)},$$

such that $s \in \mathbf{S}^j_{j'}$ if and only if there are exactly j' many $x \in \mathbf{B}_X$ for which $E_3^{-1}(s, x)$ is defined in history. $|\mathbf{S}^j_{j'}| > 0$ actually implies,

$$i_1 + j - j' \leq i.$$

$\mathbf{OP}_{XS}$ be the set of (X_j, S_j) tuples present in history. We have,

$$|\mathbf{OP}_{XS}| = i.$$

$\mathbf{OP}_{ST}$ is defined as before. Let, us denote $|\mathbf{X}^k \cap \mathbf{B}_X|$ as r_k. Note,

$$\sum_{k=1}^{i} r_k = |\mathbf{B}_X| = i_1 \text{ and } \sum_{k=1}^{i} k r_k \geq i_1.$$

Lemma 11. *If $R \in \mathbf{R}^{i_1}$, then for $k = 0, \cdots, i$ and $j = 0, \cdots, i$ we have*

$$\Pr[(X^E, S^E, T^E) = (X, S, T)] = \frac{1}{|\mathbf{Y}| - i_1} \times \frac{1}{|\mathbf{Y}| - k} \times \frac{1}{|\mathbf{Y}| - j},$$

when $(X, S, T) \in ((\mathbf{X}^k \setminus \mathbf{B}_X) \times \mathbf{S}^j \times \mathbf{Y}) \setminus ((\mathbf{OP}_{XS} \times \mathbf{Y}) \cup (\mathbf{Y} \times \mathbf{OP}_{ST}))$.

Lemma 12. *If $R \in \mathbf{R}^{i_1}$, then for $j = 0, \cdots, i$ and $j' = 0, \cdots, \min(i_1, j)$ we have*

$$\Pr[(X^P, S^P, T^P) = (X, S, T)] = \frac{1}{|\mathbf{Y}| - i_1 - j + j'} \times \frac{1}{|\mathbf{Y}|^2 - i},$$

when $(X, S, T) \in ((\mathbf{Y} \setminus \mathbf{B}_X) \times \mathbf{S}^j_{j'} \times \mathbf{Y}) \setminus ((\mathbf{OP}_{XS} \times \mathbf{Y}) \cup (\mathbf{Y} \times \mathbf{OP}_{ST}))$.

Proof. For a TYPE I query sequence the simulator response is decided by simulators behavior on $E_1(R, L)$ query. (S, T) values return by the simulator is actually direct output of P. Hence, the distribution of (S^E, T^E) is independent of internal random choices of the simulator. If we fix S then the distribution of X^E is actually uniform over the values for which $E_1^{-1}(R, .)$ and $E_3(S, .)$ is not defined. (Note, we can actually drop the $\mathsf{ConflictE}_2$ call inside $\mathsf{Check}(L, R, X)$ function when it is being called from E_1). □

Note, both (X^E, S^E, T^E) and (X^P, S^P, T^P) has non-zero probabilities over same set of points (this is a consequence of our simulator being always consistent to P, and never aborting), although they are not the same. $\mathsf{Adv}^{\mathcal{D}'}_{i+1}$ is nothing but half of sum of the probability differences of the two distributions over all points. The distribution of $(X^E, S^E, T^)$ does not depend on j', where as the distribution of (X^P, S^P, T^P) does not depend on k. To give an upper bound for the sum of probability differences we divide the total probability space in four parts. Then, we give an upper bound for each part separately.

1. Δ_{00} is the sum of probability differences for the points, where $j = 0$ and $k = 0$. Formally,

$$\Delta_{00} = \sum_{\substack{(X,S,T)\in \\ (\mathbf{Y}\setminus\mathbf{X})\times(\mathbf{Y}\setminus\mathbf{S})\times\mathbf{Y}}} \Big| \Pr[(X^E, S^E, T^E) = (X, S, T)] - \Pr[(X^P, S^P, T^P) = (X, S, T)] \Big|$$

2. Δ_{01} is the sum of probability differences for the points where, $j = 0$ and $k \geq 1$. Formally,

$$\Delta_{01} = \sum_{k=1}^{i} \sum_{\substack{(X,S,T)\in \\ (\mathbf{X}^k\setminus\mathbf{B}_X)\times(\mathbf{Y}\setminus\mathbf{S})\times\mathbf{Y}}} \Big| \Pr[(X^E, S^E, T^E) = (X, S, T)] - \Pr[(X^P, S^P, T^P) = (X, S, T)] \Big|$$

3. Δ_{10} is the sum of probability differences for the points where, $j \geq 1$ and $k = 0$. Formally,

$$\Delta_{01} = \sum_{j=1}^{i} \sum_{j'=0}^{\min(i_1,j)} \sum_{\substack{(X,S,T)\in \\ (\mathbf{Y}\setminus\mathbf{X})\times((\mathbf{S}^j_{j'}\times\mathbf{Y})\setminus\mathbf{OP}_{ST})}} \Big| \Pr[(X^E, S^E, T^E) = (X, S, T)] - \Pr[(X^P, S^P, T^P) = (X, S, T)] \Big|$$

4. Δ_{11} is the sum of probability differences for the points where, $j \geq 1$ and $k \geq 1$. Formally,

$$\Delta_{11} = \sum_{k=1}^{i} \sum_{j=1}^{i} \sum_{j'=0}^{\min(i_1,j)} \sum_{\substack{(X,S,T)\in \\ ((\mathbf{X}^k\setminus\mathbf{B}_X)\times\mathbf{S}^j_{j'}\times\mathbf{Y}) \\ \setminus((\mathbf{OP}_{XS}\times\mathbf{Y})\cup(\mathbf{Y}\times\mathbf{OP}_{ST}))}} \Big| \Pr[(X^E, S^E, T^E) = (X, S, T)] - \Pr[(X^P, S^P, T^P) = (X, S, T)] \Big|$$

Below, we state the upper bounds for Δ_{ij}'s. In section F.2 we give a detailed analysis.

1. $\Delta_{00} \leq \frac{2i}{|\mathbf{Y}|^2}$
2. $\Delta_{01} \leq \frac{4i}{|\mathbf{Y}|^2}$
3. $\Delta_{10} \leq \frac{4i}{|\mathbf{Y}|^2} + \frac{10i^2}{|\mathbf{Y}|^3}$
4. $\Delta_{11} \leq \frac{40i^2}{|\mathbf{Y}|^3} + \frac{8i^3}{|\mathbf{Y}|^4}$

Hence,

$$\begin{aligned}\mathsf{Adv}^{\mathcal{D}'}_{i+1} &= \frac{1}{2}(\Delta_{00} + \Delta_{01} + \Delta_{10} + \Delta_{11}) \\ &\leq \frac{5i}{|\mathbf{Y}|^2} + \frac{25i^2}{|\mathbf{Y}|^3} + \frac{4i^3}{|\mathbf{Y}|^4}\end{aligned}$$

F.2 Upper bound for Δ_{ij}'s

Upper bound for Δ_{00}

If $(X, S, T) \in (\mathbf{Y} \setminus \mathbf{X}) \times (\mathbf{Y} \setminus \mathbf{S}) \times \mathbf{Y}$, we have

$$\begin{aligned}&\big|\Pr[(X^E, S^E, T^E) = (X, S, T)] - \Pr[(X^P, S^P, T^P) = (X, S, T)]\big| \\ &= \big|\frac{1}{|\mathbf{Y}| - i_1} \times \frac{1}{|\mathbf{Y}|^2} - \frac{1}{|\mathbf{Y}| - i_1} \times \frac{1}{|\mathbf{Y}|^2 - i}\big| = \frac{i}{|\mathbf{Y}|^2(|\mathbf{Y}| - i_1)(|\mathbf{Y}|^2 - i)}.\end{aligned}$$

Hence,

$$\begin{aligned}\Delta_{00} &= \frac{(|\mathbf{Y}| - |\mathbf{X}|)(|\mathbf{Y}| - |\mathbf{S}|)|\mathbf{Y}| \times i}{|\mathbf{Y}|^2(|\mathbf{Y}| - i_1)(|\mathbf{Y}|^2 - i)} \\ &= \frac{i}{|\mathbf{Y}|^2 - i} \times \frac{|\mathbf{Y}| - |\mathbf{S}|}{|\mathbf{Y}|} \times \frac{|\mathbf{Y}| - |\mathbf{X}|}{|\mathbf{Y}| - i_1} \\ &\leq \frac{i}{|\mathbf{Y}|^2 - i} \qquad (As,\ i_1 = |\mathbf{B}_X| \leq |\mathbf{X}|) \\ &\leq \frac{2i}{|\mathbf{Y}|^2}\end{aligned}$$

Upper bound for Δ_{01}

If $(X, S, T) \in (\mathbf{X}^k \setminus \mathbf{B}_X) \times (\mathbf{Y} \setminus \mathbf{S}) \times \mathbf{Y}$, we have

$$\begin{aligned}&\big|\Pr[(X^E, S^E, T^E) = (X, S, T)] - \Pr[(X^P, S^P, T^P) = (X, S, T)]\big| \\ &= \big|\frac{1}{|\mathbf{Y}| - i_1} \times \frac{1}{|\mathbf{Y}| - k} \times \frac{1}{|\mathbf{Y}|} - \frac{1}{|\mathbf{Y}| - i_1} \times \frac{1}{|\mathbf{Y}|^2 - i}\big| \\ &= \frac{k|\mathbf{Y}| - i}{|\mathbf{Y}|(|\mathbf{Y}| - i_1)(|\mathbf{Y}| - k)(|\mathbf{Y}|^2 - i)}.\end{aligned}$$

Observe,

$$|(\mathbf{X}^k \setminus \mathbf{B}_X) \times (\mathbf{Y} \setminus \mathbf{S}) \times \mathbf{Y}| = (|\mathbf{X}^k| - r_k)(|\mathbf{Y}| - |\mathbf{S}|)|\mathbf{Y}|.$$

Hence,

$$\begin{aligned}
\Delta_{01} &= \sum_{k=1}^{i} \frac{(|\mathbf{X}^k| - r_k)(|\mathbf{Y}| - |\mathbf{S}|)|\mathbf{Y}|(k|\mathbf{Y}| - i)}{|\mathbf{Y}|(|\mathbf{Y}| - i_1)(|\mathbf{Y}| - k)(|\mathbf{Y}|^2 - i)} \\
&\leq \frac{(|\mathbf{Y}| - |\mathbf{S}|)}{(|\mathbf{Y}| - i_1)(|\mathbf{Y}| - i)(|\mathbf{Y}|^2 - i)} \times \sum_{k=1}^{i}(|\mathbf{X}^k| - r_k)(k|\mathbf{Y}| - i) \qquad (As,\ k \leq i) \\
&\leq \frac{i(|\mathbf{Y}| - |\mathbf{X}|)(|\mathbf{Y}| - |\mathbf{S}|))}{(|\mathbf{Y}| - i_1)(|\mathbf{Y}| - i)(|\mathbf{Y}|^2 - i)} \\
&\qquad\qquad (As,\ \sum_{k=1}^{i} k|\mathbf{X}^k| = i, \sum_{k=1}^{i} |\mathbf{X}^k| = |\mathbf{X}|, \sum_{k=1}^{i} r_k = i_1, \sum_{k=1}^{i} kr_k \geq i_1 \text{ and } |\mathbf{Y}| \geq i) \\
&\leq \frac{i|\mathbf{Y}|}{(|\mathbf{Y}| - i)(|\mathbf{Y}|^2 - i)} \qquad (As,\ i_1 \leq |\mathbf{X}|) \\
&\leq \frac{4i}{|\mathbf{Y}|^2} \qquad (As,\ i \leq |\mathbf{Y}|/2)
\end{aligned}$$

Upper bound for Δ_{10}

If $(X, S, T) \in (\mathbf{Y} \setminus \mathbf{X}) \times ((\mathbf{S}^j_{j'} \times \mathbf{Y}) \setminus \mathbf{OP}_{ST})$, we have

$$\begin{aligned}
&\big|\Pr[(X^E, S^E, T^E) = (X, S, T)] - \Pr[(X^P, S^P, T^P) = (X, S, T)]\big| \\
&= \big|\frac{1}{|\mathbf{Y}| - i_1} \times \frac{1}{|\mathbf{Y}|} \times \frac{1}{|\mathbf{Y}| - j} - \frac{1}{|\mathbf{Y}| - i_1 - j + j'} \times \frac{1}{|\mathbf{Y}|^2 - i}\big| \\
&= \big|\frac{j'|\mathbf{Y}|^2 - (i_1 j + i)|\mathbf{Y}| + i(i_1 + j - j')}{|\mathbf{Y}|(|\mathbf{Y}| - i_1)(|\mathbf{Y}| - j)(|\mathbf{Y}| - i_1 - j + j')(|\mathbf{Y}|^2 - i)}\big| \\
&\leq \frac{j'|\mathbf{Y}|^2 + (i_1 j + i)|\mathbf{Y}| + i(i_1 + j - j')}{|\mathbf{Y}|(|\mathbf{Y}| - i_1)(|\mathbf{Y}| - j)(|\mathbf{Y}| - i)(|\mathbf{Y}|^2 - i)} \qquad (As,\ j' \leq \min(i_1, j) \text{ and } i_1 + j - j' \leq i).
\end{aligned}$$

Also,

$$|(\mathbf{Y} \setminus \mathbf{X}) \times ((\mathbf{S}^j_{j'} \times \mathbf{Y}) \setminus \mathbf{OP}_{ST})| = (|\mathbf{Y}| - |\mathbf{X}|)|\mathbf{S}^j_{j'}|(|\mathbf{Y}| - j).$$

Hence,

$$\begin{aligned}
\Delta_{10} &\leq \sum_{j=1}^{i} \sum_{j'=0}^{\min(i_1, j)} \frac{(|\mathbf{Y}| - |\mathbf{X}|)|\mathbf{S}^j_{j'}|(j'|\mathbf{Y}|^2 + (i_1 j + i)|\mathbf{Y}| + i(i_1 + j - j'))}{|\mathbf{Y}|(|\mathbf{Y}| - i_1)(|\mathbf{Y}| - i)(|\mathbf{Y}|^2 - i)} \\
&\leq \sum_{j=1}^{i} \frac{(|\mathbf{Y}| - |\mathbf{X}|)(j|\mathbf{S}^j||\mathbf{Y}|^2 + (i_1 j|\mathbf{S}^j| + i|\mathbf{S}^j|)|\mathbf{Y}| + (ii_1 + j)|\mathbf{S}^j|)}{|\mathbf{Y}|(|\mathbf{Y}| - i_1)(|\mathbf{Y}| - i)(|\mathbf{Y}|^2 - i)} \\
&\qquad\qquad (As,\ j' \leq j \text{ and } \sum_{j'=0}^{\min(j, i_1)} |\mathbf{S}^j_{j'}| = |\mathbf{S}^j|) \\
&\leq \frac{i|\mathbf{Y}|^2 + (i_1 + |\mathbf{S}|)i|\mathbf{Y}| + i(i_1|\mathbf{S}| + 1)}{|\mathbf{Y}|(|\mathbf{Y}| - i)(|\mathbf{Y}|^2 - i)} \qquad (As,\ \sum_{j=1}^{i} |\mathbf{S}^j| = |\mathbf{S}|, \sum_{j=1}^{i} j|\mathbf{S}^j| = i \text{ and } i_1 \leq |\mathbf{X}|) \\
&\leq \frac{i|\mathbf{Y}|^2 + 2i^2|\mathbf{Y}| + (i^3 + i^2)}{|\mathbf{Y}|(|\mathbf{Y}| - i)(|\mathbf{Y}|^2 - i)} \qquad (As,\ i_1 \leq i \text{ and } |\mathbf{S}| \leq i) \\
&\leq \frac{4i}{|\mathbf{Y}|^2} + \frac{10i^2}{|\mathbf{Y}|^3} \qquad (As,\ i \leq |\mathbf{Y}|/2 - 1)
\end{aligned}$$

Upper bound for Δ_{11}

If $(X,S,T) \in ((\mathbf{Y} \setminus \mathbf{B}_X) \times \mathbf{S}^j_{j'} \times \mathbf{Y}) \setminus ((\mathbf{OP}_{XS} \times \mathbf{Y}) \cup (\mathbf{Y} \times \mathbf{OP}_{ST}))$, we have

$$\begin{aligned}
&\big|\Pr[(X^E,S^E,T^E)=(X,S,T)] - \Pr[(X^P,S^P,T^P)=(X,S,T)]\big| \\
&= \big|\frac{1}{|\mathbf{Y}|-i_1} \times \frac{1}{|\mathbf{Y}|-k} \times \frac{1}{|\mathbf{Y}|-j} - \frac{1}{|\mathbf{Y}|-i_1-j+j'} \times \frac{1}{|\mathbf{Y}|^2-i}\big| \\
&= \big|\frac{(k+j')|\mathbf{Y}|^2 - (i+i_1k+kj+i_1j)|\mathbf{Y}| + (ii_1+ij-ij'+i_1kj)}{(|\mathbf{Y}|-i_1)(|\mathbf{Y}|-k)(|\mathbf{Y}|-j)(|\mathbf{Y}|-i_1-j+j')(|\mathbf{Y}|^2-i)}\big| \\
&\le \frac{(k+j')|\mathbf{Y}|^2 + (i+i_1k+kj+i_1j)|\mathbf{Y}| + (ii_1+ij-ij'+i_1kj)}{(|\mathbf{Y}|-i_1)(|\mathbf{Y}|-k)(|\mathbf{Y}|-j)(|\mathbf{Y}|-i)(|\mathbf{Y}|^2-i)} \\
&\qquad (As,\ j' \le \min(i_1,j)\ and\ i_1+j-j' \le i).
\end{aligned}$$

Also, note

$$\begin{aligned}
\big|((\mathbf{X}^k \setminus \mathbf{B}_X) \times \mathbf{S}^j_{j'} \times \mathbf{Y}) \setminus ((\mathbf{OP}_{XS} \times \mathbf{Y}) \cup (\mathbf{Y} \times \mathbf{OP}_{ST}))\big| &\le \big|((\mathbf{X}^k \setminus \mathbf{B}_X) \times \mathbf{S}^j_{j'} \times \mathbf{Y}) \setminus (\mathbf{Y} \times \mathbf{OP}_{ST})\big| \\
&= (|\mathbf{X}^k| - r_k)|\mathbf{S}^j_{j'}|(|\mathbf{Y}|-j).
\end{aligned}$$

Hence,

$$\begin{aligned}
\Delta_{11} &\le \sum_{k=1}^{i}\sum_{j=1}^{i}\sum_{j'=0}^{\min(i_1,j)} (|\mathbf{X}^k| - r_k)|\mathbf{S}^j_{j'}|(|\mathbf{Y}|-j) \\
&\qquad \times \frac{(k+j')|\mathbf{Y}|^2 + (i+i_1k+kj+i_1j)|\mathbf{Y}| + (ii_1+ij-ij'+i_1kj)}{(|\mathbf{Y}|-i_1)(|\mathbf{Y}|-k)(|\mathbf{Y}|-j)(|\mathbf{Y}|-i)(|\mathbf{Y}|^2-i)} \\
&\le \sum_{k=1}^{i}\sum_{j=1}^{i}(|\mathbf{X}^k| - r_k) \times \frac{(k|\mathbf{S}^j| + j|\mathbf{S}^j|)|\mathbf{Y}|^2 + (i+i_1k+kj+i_1j)|\mathbf{S}^j||\mathbf{Y}| + (ii_1+ij+i_1kj)|\mathbf{S}^j|}{(|\mathbf{Y}|-i_1)(|\mathbf{Y}|-k)(|\mathbf{Y}|-i)(|\mathbf{Y}|^2-i)} \\
&\qquad (As,\ j' \le j\ and\ \sum_{j'=0}^{\min(i_1,j)} |\mathbf{S}^j_{j'}| = |\mathbf{S}^j|) \\
&= \sum_{k=1}^{i}(|\mathbf{X}^k| - r_k) \times \frac{(k|\mathbf{S}|+i)|\mathbf{Y}|^2 + (i|\mathbf{S}| + i_1k|\mathbf{S}| + ki + i_1i)|\mathbf{Y}| + (ii_1|\mathbf{S}| + i^2 + ii_1k)}{(|\mathbf{Y}|-i_1)(|\mathbf{Y}|-i)^2(|\mathbf{Y}|^2-i)} \\
&\qquad (As,\ k \le i, \sum_{j=1}^{i}|\mathbf{S}^j| = |\mathbf{S}|\ and\ \sum_{j=1}^{i} j|\mathbf{S}^j| = i) \\
&\le \frac{i(|\mathbf{S}|+|\mathbf{X}|)|\mathbf{Y}|^2 + (i|\mathbf{X}||\mathbf{S}| + i^2 + ii_1|\mathbf{X}|)|\mathbf{Y}| + ii_1|\mathbf{S}||\mathbf{X}| + i^2|\mathbf{X}|}{(|\mathbf{Y}|-i_1)(|\mathbf{Y}|-i)^2(|\mathbf{Y}|^2-i)} \\
&\qquad (As,\ \sum_{k=1}^{i}|\mathbf{X}^k| = |\mathbf{X}|\ and\ \sum_{k=1}^{i} k|\mathbf{X}^k| = i) \\
&\le \frac{2i^2|\mathbf{Y}|^2 + (2i^3+i^2)|\mathbf{Y}| + (i^4+i^3)}{(|\mathbf{Y}|-i)^3(|\mathbf{Y}|^2-i)} \qquad (As,\ |\mathbf{S}| \le i, |\mathbf{X}| \le i\ and\ i_1 \le i) \\
&\le \frac{40i^2}{|\mathbf{Y}|^3} + \frac{8i^3}{|\mathbf{Y}|^4} \qquad (As,\ i \le |\mathbf{Y}|/2-1)
\end{aligned}$$