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
avradip.mandal@uni.lu
[2] University of Versailles, France
jacques.patarin@uvsq.fr
[3] ANSSI, Paris, France
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.* [29] and Dodis *et al.* [12] 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.* [4].

Keywords: indifferentiability, correlation intractability, Feistel construction.

1 Introduction

Indifferentiability. Indifferentiability has been introduced by Maurer *et al.* [22] 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 [7] 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

R. Cramer (Ed.): TCC 2012, LNCS 7194, pp. 285–302, 2012.

or an ideal block cipher. 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 $\mathcal{C}^{\boldsymbol{F}}$.[1] 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.* [7] showed that a number of variants of the Merkle-Damgård construction, used with an ideal cipher in Davies-Meyer mode, 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 [18] which triggered a lot of subsequent work [20,23,24,28], 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 [20], 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 [25] 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 [11] 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 superlogarithmic 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.

[1] It was recently pointed out that this composition theorem only holds for cryptosystems whose security is defined by so called *single-stage games* [26].

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.[2] 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 [10] 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 [9]), they were not able to conclude in the general setting. Coron, Patarin, and Seurin [9] 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 [17] later found a distinguishing attack against the simulator given in [9], therefore invalidating the indifferentiability proof.[3] Only recently, Holenstein *et al.* [14] 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 [27] but had some gaps.

Public Indifferentiability. Yoneyama *et al.* [29] and Dodis *et al.* [12] 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 [2], probabilistic FDH [6], PSS [3]..., 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. [29,12] were both concerned with indifferentiability from a 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.* [4] 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

[2] 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.

[3] 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.

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 [12,29] 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 [13] and subsequent variants [15,5]) 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 [14]).

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 [9] 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 [16].

On a slightly different topic, we also analyze the Feistel-like domain extension construction for ideal ciphers proposed by Coron *et al.* [8] and show that in the seq-indifferentiability model one can obtain a security bound beyond the birthday barrier. See the full version of the paper [19].

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 [4, 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 [1] 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 [1] establishes[4] 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.

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 $\Psi_k^{\boldsymbol{F}} = \Psi_k^{(\boldsymbol{F}_1,\ldots,\boldsymbol{F}_k)}$.

[4] Strictly speaking, the result is proven in [1] for one-sided functions and permutations, but the proof can be straightforwardly adapted to two-sided primitives.

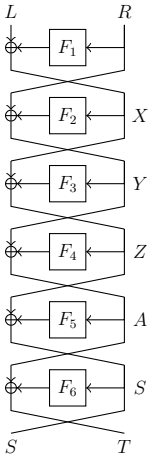

Fig. 1. Notations used for the 6-round Feistel construction

2.2 Sequential Indifferentiability

Indifferentiability was originally formulated within the formalism of *random systems* [21]. We adopt here the simpler formulation using interactive Turing machines as in [7]. We first recall the classical definition of indifferentiability [22]. 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.

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.*

Full indifferentiability obviously implies seq-indifferentiability. Yoneyama *et al.* [29] and Dodis *et al.* [12] 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 [12], the composition theorem of [22] 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*[5] *ideal primitives* $\boldsymbol{G}$, thus making seq- and pub-indifferentiability equivalent notions in that case.

[5] 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.

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. See the full version of the paper [19]. □

Ristenpart[6] observed that the above theorem does not hold (at least in the computational setting) when $\boldsymbol{G}$ is stateful. This is explained in the full version of the paper [19]. 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 [7] that it is not indifferentiable from a random oracle (a consequence of length-extension attacks), and in [12] that it is pub-indifferentiable from a random oracle.

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 [9], 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 the full version of the paper [19].

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

[6] Personal communication.

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 [9], 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.

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`

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)
    if $i = 4$ **then**
      for all $Y \in F_3$ **do**
        `CompleteForward`(Y, U)
  return $F_i(U)$

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

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

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 2.

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

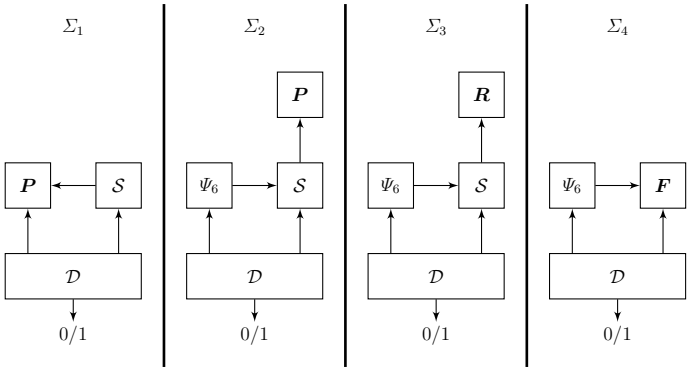

Fig. 2. Systems used in the seq-indifferentiability proof

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 [1]. □

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 20, 21, 30, and 31 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 20, 21, 30 or 31 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.[7] 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 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 $\boldsymbol{P}$, 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 bound the probability that `Bad` happens in Σ_2 or Σ_3.

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. See the full version of the paper [19]. □

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 Σ_2, then for any input $x \in \{0,1\} \times \{0,1\}^{2n}$ computable with respect to the final history of the simulator $\mathcal{H}$, $\Psi_6^{\mathcal{H}}(x) = \boldsymbol{P}(x)$.*

[7] In previous work on indifferentiability of the Feistel construction [9,27], in such a case the simulator aborted. It does not change much since, as we will prove, this happens only with negligible probability.

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 `CompleteForward`(Y, Z) or `CompleteBackward`(Y, Z) during the execution of the simulator. With respect to the history $\mathcal{H}'$ just after the completion of `CompleteForward`(Y, Z) or `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).[8] 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.*

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 20, 21, 30, and 31 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:*

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

[8] 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. 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.* [12, 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 [4]. 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:

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

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. See the full version of the paper [19]. □

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.* [14] 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.

Remark 3. According to Theorem 3, sequential indifferentiability implies correlation intractability. However correlation intractability does not necessarily imply sequential indifferentiability. In the full version of the paper [19] we provide a simple counter-example separating the two notions.

Implications for Chosen-Key and Known-Key Attacks on Block Ciphers. Knudsen and Rijmen [16] have introduced so-called known-key attacks on block ciphers. We discuss the implications of our results regarding this attack model in the full version of the paper [19].

References

1. Bellare, M., Ristenpart, T.: Multi-Property-Preserving Hash Domain Extension and the EMD Transform. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 299–314. Springer, Heidelberg (2006)
2. Bellare, M., Rogaway, P.: Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In: ACM Conference on Computer and Communications Security, pp. 62–73 (1993)
3. Bellare, M., Rogaway, P.: The Exact Security of Digital Signatures - How to Sign with RSA and Rabin. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 399–416. Springer, Heidelberg (1996)
4. Canetti, R., Goldreich, O., Halevi, S.: The Random Oracle Methodology. In: Symposium on Theory of Computing - STOC 1998, pp. 209–218. ACM (1998), revisited (Preliminary Version); Full version, `http://arxiv.org/abs/cs.CR/0010019`
5. Chevallier-Mames, B., Phan, D.H., Pointcheval, D.: Optimal Asymmetric Encryption and Signature Paddings. In: Ioannidis, J., Keromytis, A.D., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 254–268. Springer, Heidelberg (2005)
6. Coron, J.-S.: Optimal Security Proofs for PSS and Other Signature Schemes. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 272–287. Springer, Heidelberg (2002)
7. Coron, J.-S., Dodis, Y., Malinaud, C., Puniya, P.: Merkle-Damgård Revisited: How to Construct a Hash Function. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 430–448. Springer, Heidelberg (2005)
8. Coron, J.-S., Dodis, Y., Mandal, A., Seurin, Y.: A Domain Extender for the Ideal Cipher. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 273–289. Springer, Heidelberg (2010)
9. Coron, J.-S., Patarin, J., Seurin, Y.: The Random Oracle Model and the Ideal Cipher Model Are Equivalent. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 1–20. Springer, Heidelberg (2008)
10. Dodis, Y., Puniya, P.: On the Relation Between the Ideal Cipher and the Random Oracle Models. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 184–206. Springer, Heidelberg (2006)
11. Dodis, Y., Puniya, P.: Feistel Networks Made Public, and Applications. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 534–554. Springer, Heidelberg (2007)
12. Dodis, Y., Ristenpart, T., Shrimpton, T.: Salvaging Merkle-Damgård for Practical Applications. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 371–388. Springer, Heidelberg (2009)
13. Granboulan, L.: Short Signatures in the Random Oracle Model. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 364–378. Springer, Heidelberg (2002)
14. Holenstein, T., Künzler, R., Tessaro, S.: The Equivalence of the Random Oracle Model and the Ideal Cipher Model. In: Fortnow, L., Vadhan, S.P. (eds.) Symposium on Theory of Computing - STOC 2011, pp. 89–98. ACM (2011) (revisited)
15. Katz, J., Wang, N.: Efficiency improvements for signature schemes with tight security reductions. In: Jajodia, S., Atluri, V., Jaeger, T. (eds.) ACM Conference on Computer and Communications Security, pp. 155–164. ACM (2003)
16. Knudsen, L.R., Rijmen, V.: Known-Key Distinguishers for Some Block Ciphers. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 315–324. Springer, Heidelberg (2007)

17. Künzler, R.: Are the random oracle and the ideal cipher models equivalent? Master's thesis, ETH Zurich, Switzerland (2009)
18. Luby, M., Rackoff, C.: How to Construct Pseudorandom Permutations from Pseudorandom Functions. SIAM Journal on Computing 17(2), 373–386 (1988)
19. Mandal, A., Patarin, J., Seurin, Y.: On the Public Indifferentiability and Correlation Intractability of the 6-Round Feistel Construction. ePrint Archive Report 2011/496 (2011), http://eprint.iacr.org/2011/496.pdf
20. Maurer, U.M.: A Simplified and Generalized Treatment of Luby-Rackoff Pseudorandom Permutation Generators. In: Rueppel, R.A. (ed.) EUROCRYPT 1992. LNCS, vol. 658, pp. 239–255. Springer, Heidelberg (1993)
21. Maurer, U.M.: Indistinguishability of Random Systems. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 110–132. Springer, Heidelberg (2002)
22. Maurer, U., Renner, R., Holenstein, C.: Indifferentiability, Impossibility Results on Reductions, and Applications to the Random Oracle Methodology. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 21–39. Springer, Heidelberg (2004)
23. Naor, M., Reingold, O.: On the Construction of Pseudorandom Permutations: Luby-Rackoff Revisited. Journal of Cryptology 12(1), 29–66 (1999)
24. Patarin, J.: Security of Random Feistel Schemes with 5 or More Rounds. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 106–122. Springer, Heidelberg (2004)
25. Ramzan, Z., Reyzin, L.: On the Round Security of Symmetric-Key Cryptographic Primitives. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 376–393. Springer, Heidelberg (2000)
26. Ristenpart, T., Shacham, H., Shrimpton, T.: Careful with Composition: Limitations of the Indifferentiability Framework. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 487–506. Springer, Heidelberg (2011)
27. Seurin, Y.: Primitives et protocoles cryptographiques à sécurité prouvée. PhD thesis, Université de Versailles Saint-Quentin-en-Yvelines, France (2009)
28. Vaudenay, S.: Decorrelation: A Theory for Block Cipher Security. Journal of Cryptology 16(4), 249–286 (2003)
29. Yoneyama, K., Miyagawa, S., Ohta, K.: Leaky Random Oracle. IEICE Transactions 92-A(8), 1795–1807 (2009)