Indifferentiability of Confusion-Diffusion Networks

Yevgeniy Dodis[1(✉)], Martijn Stam[2], John Steinberger[3], and Tianren Liu[4]

[1] Courant Institute, New York University, New York, USA
dodis@cs.nyu.edu
[2] Department of Computer Science, University of Bristol, Bristol, UK
csxms@bristol.ac.uk
[3] Institute for Interdisciplinary Information Sciences,
Tsinghua University, Beijing, China
jpsteinb@gmail.com
[4] MIT, Cambridge, USA
liutianren@gmail.com

Abstract. We show the first positive results for the indifferentiability security of the confusion-diffusion networks (which are extensively used in the design of block ciphers and hash functions). In particular, our result shows that a constant number of confusion-diffusion rounds is sufficient to extend the domain of a public random permutation.

1 Introduction

In this work we simultaneously address the following two questions:

- **Question 1:** secure domain extension of a *public* random permutation.
- **Question 2:** theoretical soundness of Shannon's (or Feistel's) confusion-diffusion paradigm.

Domain Extension of RPs. The question of domain extension of various cryptographic primitives, such as encryption, signatures, message authentication codes, pseudorandom functions (PRFs), pseudorandom permutations (PRPs), etc., is one of the fundamental questions in cryptography.

In this paper we address a similar question for a *public* random permutation. Namely, given one (or a constant number of) n-bit random permutation(s) $P : \{0,1\}^n \to \{0,1\}^n$, and a number $w \geq 2$, build a wn-bit random permutation $\mathcal{Z} : \{0,1\}^{wn} \to \{0,1\}^{wn}$. This question is clearly natural and interesting it is own right, but also seems extremely relevant in practice. Indeed, the random

Y. Dodis—Partially supported by gifts from VMware Labs and Google, and NSF grants 1319051, 1314568, 1065288, 1017471.

J. Steinberger—Supported by National Basic Research Program of China Grant 2011CBA00300, 2011CBA00301, the National Natural Science Foundation of China Grant 61033001, 61361136003, and by the China Ministry of Education grant number 20121088050.

M. Fischlin and J.-S. Coron (Eds.): EUROCRYPT 2016, Part II, LNCS 9666, pp. 679–704, 2016.
DOI: 10.1007/978-3-662-49896-5_24

permutation model (RPM) has recently received a lot of attention [2,13,26,29], starting to "compete with" and perhaps even "overtake" the more well known random oracle model (ROM) and the ideal cipher model (ICM). Aside from elegance, one of the reasons for this renewed attention comes from the fact that one can abstract the design of both the block-cipher standard AES and the new SHA-3 standard Keccak as being in the RPM. Namely, AES can be viewed as a 10-round *key-alternating* cipher applied to a concrete ("random-looking") permutation, while SHA-3 can be viewed as applying a "sponge" mode of operation [2] to a similarly "random-looking" permutation. In fact, in his invited talk at Eurocrypt'13, the designer of both AES and SHA-3 Joan Daemen claimed that the RPM is much closer to the existing practice of designing hash functions and block ciphers than either the ROM or ICM, challenging the cryptographic community to switch to the RPM.

Of course, one must now build those "random looking permutations" $\mathcal{Z}$ on relatively large domains (perhaps from 128 bits, like AES-128, to 1600 bits, like Keccak, or even longer). In practice, we have two well-known methods for accomplishing such a goal. The first method is based on applying several rounds of the Feistel network to some (not necessarily invertible) round functions. In our (public) setting, this method was theoretically analyzed only recently by Holenstein et al. [18] (building on an earlier work of [11]), who showed that a 14-round Feistel network is indeed sufficient for building a random permutation (RP), provided the round functions are modeled as (easily made) independent random oracles (ROs). Although very important in theory (i.e., showing the equivalence between ROM and RPM), this method does not seem to be used in practice, as it appears almost as hard,—if not *harder*,—to design "random-looking" non-invertible round functions on large domains as it is to design the desired random-looking permutation $\mathcal{Z}$.

Confusion-Diffusion Paradigm. Instead, practitioners use the second method,—the *confusion-diffusion* (CD) paradigm,[1]—which directly connects our motivating Questions 1 and 2. The idea of CD goes back to the seminal paper of Feistel [15] and even[2] back to Shannon [28]. Abstractly, one splits the input x to $\mathcal{Z}$ into several shorter blocks $x_1 \dots x_w$, and then alternates the following two steps for several rounds: (a) *Confusion*, which consists of applying some

[1] This is closely related to the substitution-permutation network (SPN) paradigm. Historically, though, the term SPN usually refers to the design of block ciphers as opposed to a single permutation, where one also XORs some key material in between successive CD rounds. To avoid confusion, we will stick with the term CD and not use the term SPN.

[2] Shannon [28] introduces "confusion" and "diffusion" into the cryptographic lexicon while Feistel [15] articulates the modern notion of a confusion-diffusion network, crediting Shannon with inspiration. There are some notable gaps between Shannon and the modern viewpoint. In particular Shannon does not seem to view confusion as a local operation, nor does he advocate repeatedly alternating steps of "confusion" and "diffusion". Instead, Shannon seems to view confusion and diffusion as globally desirable attributes of a cryptographic mixing operation.

fixed short permutations $P_1 \dots P_w$ (called *S-boxes*) to $x_1, \dots x_w$; and (b) *Diffusion*, which consists of applying some "mixing" non-cryptographic permutation $\pi(y_1 \dots y_w)$ (typically, carefully chosen linear function, sometimes also called *D-box*) to the results $y_1 \dots y_w$ of step (a).

Despite its extensive use in practice, the CD paradigm received extremely little attention from the theoretical cryptographic community.[3] A notable exception is a beautiful work of Miles and Viola [23], who only looked at the secret-key setting—where the permutations $P_1, \dots, P_w$ are secret— and also primarily considered the "weaker-than-indistinguishability" properties which can be proven about CD (and, more generally, SPN networks). In contrast, we are interested in the public setting, where the permutations P_i are modeled as RPs, and seek to examine the *indifferentiability* properties [9,21] of the CD paradigm. This leads us to the following more precise reformulation of our motivating Questions 1 and 2:

– **Main Question:** *Analyze indifferentiability of the confusion-diffusion paradigm as a way to extend the domain of a (constant number of) random permutation(s).* More precisely, for how many rounds r, and under what conditions on the D-boxes $\pi_1 \dots \pi_r$, is the r-round CD paradigm indifferentiable from an nw-bit random permutation $\mathcal{Z}$?

Before presenting our results, we make a few remarks. First, we will model the "small permutations" P_i as both random and independent. The independence assumption is crucially used in our current proofs, but does not appear necessary. Unfortunately, the proofs we have are already extremely involved, so we feel this initial simplification is justified. We notice similar abstractions are made by most other papers in the area (including the seminal Luby-Rackoff paper [20] and the indifferentiability results of [11,18]), though one hopes it might be lifted in future work.

As for modeling P_i as random, it seems inherent if we want to build a random permutation $\mathcal{Z}$; e.g., we cannot build it from "nothing" (as this implies $P \neq NP$ and more), and it seems unlikely that any weaker assumption on the P_i will work. However, it does come with an important caveat: the best security bound ε we can naturally get with this approach will certainly be $\varepsilon \gg 2^{-n}$, where n is the domain of the S-boxes P_i. In practice, however, the S-boxes use a very small value of n (e.g., $n = 8$ for the AES), partly so that S-boxes can be easily and efficiently implemented as lookup tables. With such a small value of n, however, our bounds appear "practically meaningless", irrespective of the number of queries q made by the attacker. This means that none of our results would be directly applicable to any of the "practical" permutations $\mathcal{Z}$ used in the existing hash functions and block ciphers. Still, we believe establishing "structural soundness" of the CD paradigm is an important conceptual contribution—and an overdue sanity check—even with this serious (and inherent) limitation.

[3] Of course, there is a lot of cryptanalytic work in the area whose survey is beyond the scope of this work.

1.1 Overview of Our Results

We give a sequence of results establishing the soundness of the CD paradigm as a method for domain-extension of random permutations. Our indifferentiability results include CD networks of 5, 6, 7, 9, 10 and 11 rounds. These networks achieve different security levels, different query complexities, and place different combinatorial requirements on the D-boxes as well, even within the same network. Figure 1 summarizes the main bounds achieved for each network length, up to lower-order (e.g., logarithmic) factors, and subject to various caveats to be shortly explained.

rounds	(flags)	D-boxes	ε $(w=2)$	q_S $(w=2)$	ε $(w>2)$	q_S $(w>2)$
5	(000)	arbitrary	$q^4/2^n$	q^4	$q^{2w}/2^n$	q^{w^2}
6	(100)	arbitrary	$q^2/2^n$	q^2	$q^2/2^n$	q^w
7	(110)	arbitrary	$q^2/2^n$	q	$q^2/2^n$	q
5	(000)	GF(2^n)-linear	$q^4/2^n$	q^4	1	–
6	(100)	GF(2^n)-linear	$[q^{10/3}/2^n]$	$[q^{10/3}]$	1	–
7	(110)	GF(2^n)-linear	$[q^5/2^n]$	$[q^{25/9}]$	1	–
9	(001)	GF(2^n)-linear	$q^4/2^n$	q^4	$q^{2w}/2^n$	q^{w^2}
10	(101)	GF(2^n)-linear	$[q^{10/3}/2^n]$	$[q^{10/3}]$	$[q^4/2^n]$	$[q^{2w-1/2}]$
11	(111)	GF(2^n)-linear	$[q^5/2^n]$	$[q^{25/9}]$	$[q^6/2^n]$	$[q^4]$

Fig. 1. Summary of security ε and simulator query complexity q_S (as functions of block length n, width w, and the number of distinguisher queries q) across our six main simulators with arbitrary or GF(2^n)-linear permutations. Entries in square brackets are not known, with the value inside the brackets being conjectured based on current best-known bounds. Constants and logarithmic factors are elided for simplicity. The meaning of the bit sequence next to each round number is explained in Sect. 5.

To read Fig. 1, recall that n is the block length of the S-boxes and that w is the width of the network. Moreover, q is the number of distinguisher queries, ε is the simulator's security (i.e., the indistinguishability of the real and simulated worlds), and lastly q_S is the simulator's query complexity (see Sect. 2 for definitions). The meaning of the 3-bit sequence next to each round number in the left column will be explained in Sect. 5.

The first three rows of Fig. 1 present our best bounds (i.e., assuming a "smart" choice of the D-boxes) when the D-boxes are not restricted to be GF(2^n)-linear. One can observe that in this part of the table the bounds for $w = 2$ are simply obtained by plugging in $w = 2$ to the general bounds. The best simulator here, at 7 rounds, achieves "birthday" security of $q^2/2^n$ and an essentially optimal query complexity of q.

Concerning the first three rows of Fig. 1 there is only one caveat: for some networks (specifically those of 6 and 7 rounds, and for both $w = 2$ and $w > 2$) the bounds presume that certain D-boxes have low "conductance"—a new critical property that we introduce and elaborate on below. Such permutations are known to exist on probabilistic grounds, but so far we do not know any explicit constructions, and building explicit permutations with low conductance is indeed one of the more interesting open problems raised by our work.

The last six rows of Fig. 1 concern the case when all D-boxes in the network are required to be $\mathrm{GF}(2^n)$-linear, which is a case of interest because it aligns with most practical constructions. In this case, the simulators at width $w > 2$ and at 5, 6 and 7 rounds are not secure at all: $\varepsilon = 1$. In this regime, indeed, our simulator places certain combinatorial requirements on some of the D-boxes that are not satisfied by any $\mathrm{GF}(2^n)$-linear permutation. Fortunately, these requirements can be relaxed by using four more (i.e., 9, 10 or 11) rounds, so that $\mathrm{GF}(2^n)$-linear D-boxes become possible again for those round numbers at $w > 2$.

Unfortunately, except for the bounds at 5 rounds for $w = 2$ and the (rather poor) bounds at 9 rounds for $w > 2$, remaining bounds in the $\mathrm{GF}(2^n)$-linear section of the table are speculative, this being again related to "conductance"—specifically, the issue is that the lowest possible conductance of a $\mathrm{GF}(2^n)$-linear permutations is not currently known. We show some nontrivial lower bounds on the conductance of *generic* $\mathrm{GF}(2^n)$-linear permutations in the full version [8], but we have no similar nontrivial upper bounds! The conjectured bounds contained in Fig. 1 are obtained by using our lower bounds as a guess for the actual lowest possible conductance, and by rounding up some inconvenient exponents.[4] The "true" values in the lower part of the table may well turn out to be *lower* than our conjectured values (if *non-generic* $\mathrm{GF}(2^n)$-linear permutations with low conductance turn out to exist) or *higher* (if even better lower bounds on the conductance of all $\mathrm{GF}(2^n)$-linear permutations are proved).

Still in the same part of the table, one can also note that going from 6 to 7 rounds or from 10 to 11 rounds entails a *decrease* in security. In our simulator, indeed, the actual purpose of adding the extra round (from 6 to 7 or from 10 to 11) is to improve the query complexity!

Combinatorial Properties of D-boxes. We note that our general theorem statement (found in Sect. 4) makes no distinction between linear and nonlinear cases; it simply expresses the security, query complexity and runtime of the simulator as a function of various combinatorial metrics (such as conductance mentioned above) of the diffusion permutations that are present in the network. Different metrics matter for different D-boxes, depending on their position in the networks, leading to a subtle (but also modular and fine-grained) result. In our opinion, the identification of precise (and distinct!) combinatorial requirements for each D-box of the network is one of the interesting contributions of this work,

[4] If trivial upper bounds on conductance are applied instead, the rows for the 6- and 7-round networks with $w = 2$ in the second part of the table become the same as the row for the 5-round network in that half, while the rows for the 10- and 11-round networks become the same as the row for the 9-round network.

as it potentially allows future constructions to optimize the design of D-boxes layer by layer, with respect to the specific metrics targeted by each layer.

Moreover, some of our metrics entirely disappear when more rounds are added (specifically, by going from 5, 6, 7 rounds to 9, 10, 11 rounds respectively), leading to a relaxation of the conditions on the D-boxes at a larger number of rounds. For example, as already mentioned, security cannot be achieved at 5, 6 or 7 rounds with $w > 2$ and with $\mathrm{GF}(2^n)$-linear D-boxes, but can (unconditionally) be achieved by adding 4 more rounds to each of these networks, because a certain metric that no $\mathrm{GF}(2^n)$-linear permutation satisfies at $w > 2$ is no longer needed after the addition of the extra four rounds. In fact, although our table for general D-boxes (first 3 rows) in Fig. 1 does not include networks of 9, 10 and 11 rounds (since these networks would have the same security and simulator efficiency as the included networks for 5, 6 and 7 rounds, respectively) such networks are nonetheless considered by our main result, and might indeed be interesting from the point of view of increased efficiency. Namely, the weakened requirements on the D-boxes could potentially make these networks more cheap/fast/space efficient than their shorter, more combinatorially demanding counterparts! (Further such options, at even more rounds than considered by our main result, are explored in the paper's full version [8]).

In all we identify four combinatorial metrics on D-boxes that are useful for our purposes, these being (and for lack of better terminology) "entry-wise randomized preimage resistance" (RPR), "entry-wise randomized collision resistance" (RCR), "conductance" and its cousin "all-but-one conductance". The full definitions for these metrics can be found in Sect. 4. The first two metrics—RPR and RCR—are relatively unsurprising for our type of proof. (Briefly, they concern experiments in which all but one of the w input wires are fixed, and the final D-box input wire is drawn at random; the probability of a certain event occuring on the output wires should be low). Moreover, there is not much "mystery" in RPR and RCR, since it happens that one can construct explicit permutations that achieve essentially optimal bounds for these metrics.[5]

Indeed, we consider conductance to be a more novel and interesting metric. (All-but-one conductance is, conceptually at least, very closely related to conductance). We expand on this key metric now.

Conductance. Briefly, conductance is a function of the number of queries q; the conductance of a permutation $\pi : \{0,1\}^{wn} \to \{0,1\}^{wn}$ at q queries is the maximum over all possible pairs of cartesian products $(U_1 \times \cdots \times U_w, V_1 \times \cdots \times V_w)$, where $U_i, V_i \subseteq \{0,1\}^n$ and $|U_i| = |V_i| = q$ for each $1 \leq i \leq w$, of the numbers of pairs $(\mathbf{x}, \mathbf{y}) \in \{0,1\}^{wn} \times \{0,1\}^{wn}$ such that

$$\pi(\mathbf{x}) = \mathbf{y} \qquad \text{and} \qquad (\mathbf{x}, \mathbf{y}) \in (U_1 \times \cdots \times U_w, V_1 \times \cdots \times V_w)$$

[5] On the other hand, an interesting research direction might be to find explicit constructions of RPR and RCR permutations that achieve higher speeds than our own naïve constructions!.

In other words, one can choose q different values on each input and on each output wire, and one counts the number of intput-output pairs $(\mathbf{x}, \mathbf{y})$ that "entirely fit" inside the induced cartesian products.

Now if one imagines the D-box π to be sandwiched between two rounds of S-boxes—as it will be in the network—the relevance of conductance to our setting can easily be guessed: U_i corresponds to the set of values that are queried outputs of the i-th S-box in the round before π, V_i corresponds to the set of values that are queried inputs of the i-th S-box in the round after π, and the conductance is an upper bound on the number of "all consistent input-output pairs that can be assembled" from these S-box queries under π's mapping. Intuitively, if the number of such pairs is low, the job of the indifferentiability simulator is easier, as it has to worry about fewer "consistent chains" that the distinguisher is trying to assemble.

It is easy to see from the definition that conductance for any permutation $\pi : \{0,1\}^{wn} \to \{0,1\}^{wn}$ lies between q and q^w. In [8] we show that a random permutation has conductance close to $wqn \approx q$ and that the conductance of a generic $\mathrm{GF}(2^n)$-linear permutation is at least $q^{2-1/(2w-1)}$, which is always super-linear in q. An already-mentioned corollary is that, and at least with respect to our current simulator, having linear D-boxes seems to cause strictly worse security than what is achievable by arbitrary (albeit currently non-constructive!) D-boxes. Since it may well turn out that low conductance is instantiable by D-boxes that are no slower than the current $\mathrm{GF}(2^n)$-linear D-boxes, our work raises, among others, the question of whether $\mathrm{GF}(2^n)$-linearity is really the right choice for D-boxes in the CD paradigm.

Summary. Overall, we show the first positive results for the indifferentiability security of the confusion-diffusion paradigm which is extensively used in the design of block ciphers and hash functions. Our result shows that a constant number of confusion-diffusion rounds is sufficient to extend the domain of a public random permutation. In the process, we reduced the indifferentiability properties of the CD network (for a variety of rounds between 5 and 11) to natural and novel combinatorial properties of the D-boxes (such as conductance), which we hope will lead to a better understanding of the confusion-diffusion networks, and will be useful and the future design and analysis of block ciphers and hash functions.

1.2 Other Related Work

The question of domain extension ideal primitives was considered by [9,22] for the setting of public random functions (ROM), and by [10] for the setting of block ciphers (ICM). While none if these domain extensions directly apply to the RPM (e.g., the result of [10] crucially relies of the existence of a key for the "small ideal cipher"), they can be composed with the series of results showing the equivalence between RPM, ICM and ROM [1,9,11–14,18] to achieve various theoretical domain extension methods in the RPM. For example, one can get a "small RO" from "small RP" [12,13], extend domain of RO [9], and apply the

14-round Feistel construction to get a large-domain RP [18] (many other combinations of prior results will suffice as well). However, all such combinations of prior work will be *much* less efficient (and elegant) than our natural construction, and, more importantly, such results will not correspond to the way random permutations are built in real life.

The domain extension of *secret-key* random permutations is well studied: examples include PEP [4], XCB [16], HCTR [32], HCH [5] and TET [17] (and even the original Feistel constructions [20,24] could be viewed as domain doubling techniques in this setting). However, it is easy to see that none of those constructions provide the indifferentiability property in the public permutation setting.

Finally, the design of public permutations is related in spirit to the area of *white-box cryptography* [6,7], with the idea to "obfuscate" key-dependent parts of the cipher and publish them as lookup tables, making the entire construction public. We refer to [3] for an excellent discussion of this big area of research, as well as a survey of many cryptanalytic efforts attacking various SPN designs with linear diffusion layers.

2 Definitions

Basic Notations. We write $[w]$ for the set of integers $\{1, \ldots, w\}$. Elements of $\{0,1\}^{wn}$ are written with bold letters such as $\mathbf{x}$, $\mathbf{y}$; the i-th n-bit block of $\mathbf{x} \in \{0,1\}^{wn}$, $1 \le i \le w$, is written $\mathbf{x}[i]$.

Confusion-Diffusion Networks. Fix integers $w, n, r \in \mathbb{N}$. Let

$$\mathcal{P} = \{P_{i,j} : (i,j) \in [r] \times [w]\}$$

be an array of rw permutations from $\{0,1\}^n$ to $\{0,1\}^n$, i.e., $P_{i,j}$ is a permutation from $\{0,1\}^n$ to $\{0,1\}^n$ for each $i \in [r]$ and each $j \in [w]$. Also let

$$\overline{\pi} = (\pi_1, \ldots, \pi_{r-1})$$

be an arbitrary sequence of $r-1$ permutations, each from $\{0,1\}^{wn}$ to $\{0,1\}^{wn}$.

Given $\mathcal{P}$ and $\mathbf{x} \in \{0,1\}^{wn}$ we let

$$P_i(\mathbf{x})$$

denote the value in $\{0,1\}^{wn}$ obtained by applying the permutations $P_{i,1}, \ldots, P_{i,w}$ blockwise to $\mathbf{x}$. In other words, $P_i : \{0,1\}^{wn} \to \{0,1\}^{wn}$ is defined by setting

$$P_i(\mathbf{x})[j] = P_{i,j}(\mathbf{x}[j])$$

for all $j \in [w]$. It is obvious that P_i is a permutation of $\{0,1\}^{wn}$.

Given $\mathcal{P}$ and $\overline{\pi}$, we define the permutation $P = P[\mathcal{P}, \overline{\pi}]$ from $\{0,1\}^{wn}$ to $\{0,1\}^{wn}$ as the composition

$$P[\mathcal{P}, \overline{\pi}] = P_r \circ \pi_{r-1} \circ \ldots \circ P_2 \circ \pi_1 \circ P_1.$$

I.e.,

$$P[\mathcal{P}, \overline{\pi}](\mathbf{x}) = P_r(\pi_{r-1}(\ldots P_2(\pi_1(P_1(\mathbf{x})))\ldots))$$

for $\mathbf{x} \in \{0,1\}^{wn}$. We call $P[\mathcal{P}, \overline{\pi}]$ the *confusion-diffusion network* built from $\mathcal{P}$ and $\overline{\pi}$. The permutations in $\mathcal{P}$ are variously called the *confusion permutations* or *S-boxes*. The permutations in $\overline{\pi}$ are variously called the *diffusion permutations* or *D-boxes*.

The values n, w and r will be called the *wire length*, the *width* and the *number of rounds* respectively.

In practice, the S-boxes are implemented by "convoluted" or "random-like" permutations while the D-boxes are implemented by "easy" (typically linear) permutations that are cryptographically weak. In our indifferentiability model, described next, the S-boxes are modeled as random permutations while the D-boxes are publically fixed parameters of the network.

Indifferentiability. Let C be a construction making calls to an ideal set of primitives $\mathcal{P}$, which we notate as $C^{\mathcal{P}}$. Let $\mathcal{Z}$ be an ideal primitive with the same interface as $C^{\mathcal{P}}$ (e.g., $\mathcal{Z}$ is a random permutation if $C^{\mathcal{P}}$ implements a permutation). Indifferentiability is meant to capture the intuitive notion that the construction $C^{\mathcal{P}}$ is "just as good" as $\mathcal{Z}$, in some precise sense. The definition involves a simulator:

Definition 1. *An (oracle) circuit C with access to a set of ideal primitives $\mathcal{P}$ is* (t_S, q_S, ε)-indifferentiable *from an ideal primitive $\mathcal{Z}$ if there exists a simulator S such that*

$$\Pr\left[D^{C^{\mathcal{P}},\mathcal{P}} = 1\right] - \Pr\left[D^{\mathcal{Z},S^{\mathcal{Z}}}\right] \leq \varepsilon$$

for every distinguisher D making at most q_0 queries to its oracles, and such that S runs in total time t_S and makes at most q_S queries to $\mathcal{Z}$. Here t_S, q_S and ε are functions of q_0.

We note that in the "real world" D has oracle access to the construction $C^{\mathcal{P}}$ as well as to the primitives $\mathcal{P}$; in the "ideal world" $C^{\mathcal{P}}$ is replaced by the ideal primitive $\mathcal{Z}$ and the ideal primitives $\mathcal{P}$ are replaced by the simulator S. Thus, S's job is to make $\mathcal{Z}$ look like $C^{\mathcal{P}}$ by inventing "answers that fit" for D's queries to the primitives in $\mathcal{P}$. For this, S requires query access to $\mathcal{Z}$ (notated as $S^{\mathcal{Z}}$); on the other hand, S does not get to see which queries D is making to $\mathcal{Z}$.

Informally, $C^{\mathcal{P}}$ is *indifferentiable* from $\mathcal{Z}$ if it is (t_S, q_S, ε)-indifferentiable for "reasonable" values of (t_S, q_S, ε). An essential composition theorem [9,21] states that any cryptosystem that is secure when implemented with $\mathcal{Z}$ remains secure if $\mathcal{Z}$ is replaced with $C^{\mathcal{P}}$, if $C^{\mathcal{P}}$ is indifferentiable from $\mathcal{Z}$. However, the class of adversaries with respect to which the cryptosystem's security is defined must be a class that is large enough to accomodate the simulator S from Definition 1. See, e.g., [25] for a dramatic example in which indifferentiability fails completely.

In our setting "$\mathcal{P}$ will be $\mathcal{P}$" (i.e., the set of ideal primitives $\mathcal{P}$ will be the set of wr independent random permutations discussed in the previous subsection), while $C^{\mathcal{P}}$ will be $P[\mathcal{P}, \overline{\pi}]$. (As explained, the diffusion permutations $\overline{\pi}$ are a fixed, publically known parameter of the construction). Consequently, $\mathcal{Z}$ (matching

$C^{\mathcal{P}}$'s syntax) will be a random permutation from $\{0,1\}^{wn}$ to $\{0,1\}^{wn}$. Like all permutation oracles, $\mathcal{Z}$ can be queried in both forward and backward directions.

3 Attack on Two-Round Confusion-Diffusion Networks

In this section we outline a simple distinguishing attack that shows confusion-diffusion networks of two rounds or less cannot be indifferentiable from a random permutation. Unfortunately we could not find a similarly general attack for networks with three rounds, which leaves open the possibility that 3- or 4-round confusion-diffusion network might already be indifferentiable.

The attack on 2-round networks requires $w \geq 2$, which is indeed a trivial requirement since if $w = 1$ then a 1-round network is already indifferentiable from a random permutation.

For concreteness we sketch the attack with $w = 2$. The confusion-diffusion network then has four S-boxes labeled $P_{i,j}$ for $(i,j) \in [2] \times [2]$ and one diffusion permutation $\pi : \{0,1\}^{2n} \to \{0,1\}^{2n}$. The S-boxes in the first round are $P_{1,j}$, $j \in [2]$, the S-boxes in the second round are $P_{2,j}$, $j \in [2]$.

We will say the distinguisher "rejects" if it believes that it is in the simulated world; "accepts" if it believes it is in the real world.

The distinguishing attack is as follows:

1. The distinguisher randomly chooses $\mathbf{x} \in \{0,1\}^{2n}$ and queries $\mathcal{Z}(\mathbf{x})$, where $\mathcal{Z} : \{0,1\}^{2n} \to \{0,1\}^{2n}$ is the random permutation, obtaining $\mathbf{y} \in \{0,1\}^{2n}$ as answer.
2. The distinguisher make the two S-box queries $P_{1,1}(\mathbf{x}[1])$ and $P_{2,1}^{-1}(\mathbf{y}[1])$ receiving answers $a \in \{0,1\}^n$ and $b \in \{0,1\}^n$ respectively.
3. If there exists no pair of values (c,d) such that $\pi(a\|c) = (b\|d)$, the distinguisher rejects.
4. If there exists a pair of values (c,d) such that $\pi(a\|c) = (b\|d)$, the distinguisher chooses any such pair, queries $P_{1,2}^{-1}(c)$ obtaining answer t, and accepts if and only if $\mathcal{Z}(\mathbf{x}[1]\|t)[1] = \mathbf{y}[1]$.

It is clear that the distinguisher always accepts in the real world. We now argue that the simulator has negligible chance of making the distinguisher accept.

It is helpful to picture the simulator as knowing the distinguisher's attack. Moreover, we can be generous to the simulator and give both $\mathbf{x}[1]$ and $\mathbf{y}[1]$ to the simulator before requesting the answers a and b from the simulator.

By choosing a and b, the simulator knows which of options 3 and 4 the distinguisher will execute, so the simulator is essentially choosing between these options when it chooses a and b.

Obviously, case 3 is no good for the simulator; moreover, the simulator has no further information on $\mathbf{x}$ and $\mathbf{y}$ besides $\mathbf{x}[1]$ and $\mathbf{y}[1]$, from which it is computationally infeasible, if $\mathcal{Z}$ is a random permutation, to locate a value t such that $\mathcal{Z}(\mathbf{x}[1]\|t)[1] = \mathbf{y}[1]$, and which rules out case 4. The simulator is therefore doomed.

4 Combinatorial Definitions

In this section (re-)define the four combinatorial metrics on diffusion permutations mentioned in the introduction, these being *entry-wise randomized preimage resistance* (RPR), *entry-wise randomized collision resistance* (RCR), *conductance* and *all-but-one-conductance*.

Properties are defined unidirectionally: π might satisfy a property while π^{-1} does not.

Given $\pi : \{0,1\}^{wn} \to \{0,1\}^{wn}$, a vector $\mathbf{x} \in \{0,1\}^{wn}$ and indices $j, j' \in [w]$, we let

$$\pi^{\mathbf{x}}_{j,j'} : \{0,1\}^n \to \{0,1\}^n$$

be the function from $\{0,1\}^n$ to $\{0,1\}^n$ obtained by restricting the i-th block of input of π, $i \neq j$, to $\mathbf{x}[i]$, by replacing $\mathbf{x}[j]$ with the input $x \in \{0,1\}^n$, and by considering only the j'-th block of output. (The value $\mathbf{x}[j]$ being, thus, immaterial to $\pi^{\mathbf{x}}_{j,j'}$, since it is replaced by the input).

Entry-Wise Randomized Preimage Resistance and Entry-Wise Randomized Collision Resistance. The *entry-wise randomized preimage resistance* (RPR) of π is denoted $\mathsf{MaxPreim}(\pi)$, and defined as

$$\mathsf{MaxPreim}(\pi) = \max_{\mathbf{x},j,h,y} |\{x \in \{0,1\}^n : \pi^{\mathbf{x}}_{j,h}(x) = y\}|$$

while the *entry-wise randomized collision resistance* (RCR) of π is denoted $\mathsf{MaxColl}(\pi)$, and defined as

$$\mathsf{MaxColl}(\pi) = \max_{\mathbf{x},\mathbf{x}',j,h} |\{x \in \{0,1\}^n : \pi^{\mathbf{x}}_{j,h}(x) = \pi^{\mathbf{x}'}_{j,h}(x)\}|$$

where the latter maximum is taken over all tuples $\mathbf{x}$, $\mathbf{x}', j, h$ such that $\mathbf{x}[j'] \neq \mathbf{x}'[j']$ for some $j' \neq j$. Then by definition

$$\Pr_x[\pi^{\mathbf{x}}_{j,h}(x) = y] \leq \frac{\mathsf{MaxPreim}(\pi)}{2^n}$$

for all $\mathbf{x} \in \{0,1\}^{wn}$, $y \in \{0,1\}^n$, and $j, h \in [w]$, where the probability is computed over a uniform choice of $x \in \{0,1\}^n$, and

$$\Pr_x[\pi^{\mathbf{x}}_{j,h}(x) = \pi^{\mathbf{x}'}_{j,h}(x)] \leq \frac{\mathsf{MaxColl}(\pi)}{2^n}$$

for all $\mathbf{x}$, $\mathbf{x}' \in \{0,1\}^{wn}$, $j, h \in [w]$, such that $\mathbf{x}[j'] \neq \mathbf{x}'[j']$ for at least one $j' \neq j$.

Small values of $\mathsf{MaxPreim}(\pi)$ and of $\mathsf{MaxColl}(\pi)$ are better. It is easy to construct permutations with $\mathsf{MaxPreim}(\pi) = 1$ (which is optimal): simply use a linear permutation $\pi : \mathrm{GF}(2^n)^w \to \mathrm{GF}(2^n)^w$ whose associated matrix (a $w \times w$ matrix with entries in $\mathrm{GF}(2^n)$) has all nonzero entries. Constructing permutations with small values of $\mathsf{MaxColl}(\pi)$ is a bit more involved; this is done in the full version [8].

An interesting research direction would be to devise faster-than-linear RPR permutations and faster-than-polynomial RCR permutations. (Our construction of RCR permutations [8] uses finite field operations).

Conductance and All-But-One Conductance. The *conductance* $\mathsf{Cond}_\pi(q)$ of a permutation π was defined in Sect. 1.1. The notion all-but-one conductance is essentially the same as conductance, except that one coordinate position in either the input or output is ignored. The all-but-one conductance of a permutation π at q queries is denoted $\mathsf{aboCond}_\pi(q)$.

Formally, given a permutation $\pi : (\{0,1\}^n)^w \to (\{0,1\}^n)^w$, we define the *all-but-one conductance* of π by

$$\mathsf{Cond}^{h,+}_\pi(q) = \max_{\substack{U_1,\ldots,U_w,V_1,\ldots,V_w \subseteq \{0,1\}^n \\ |U_1|=\cdots=|V_w|=q}} |\{(\mathbf{x},\mathbf{y}) : \mathbf{y} = \pi(\mathbf{x}), \mathbf{x}[j] \in U_j \; \forall j \in [w], \mathbf{y}[j] \in V_j \; \forall j \in [w] \backslash h\}|,$$

$$\mathsf{Cond}^{h,-}_\pi(q) = \max_{\substack{U_1,\ldots,U_w,V_1,\ldots,V_w \subseteq \{0,1\}^n \\ |U_1|=\cdots=|V_w|=q}} |\{(\mathbf{x},\mathbf{y}) : \mathbf{y} = \pi(\mathbf{x}), \mathbf{x}[j] \in U_j \; \forall j \in [w] \backslash h, \mathbf{y}[j] \in V_j \; \forall j \in [w]\}|$$

$$\mathsf{aboCond}_\pi(q) = \max\left(\max_{h\in[w]}(\mathsf{Cond}^{h,+}(\pi,q)), \max_{h\in[w]}(\mathsf{Cond}^{h,-}(\pi,q))\right)$$

(Here the first two definitions are for all $h \in [w]$, and we use $\forall$ in postfix notation). Thus the set V_h is immaterial in the definition of $\mathsf{Cond}^{h,+}$, while the set U_h is immaterial in the definition of $\mathsf{Cond}^{h,-}$.

In [8] we show that the conductance and all-but-one conductance of a random permutation π are both roughly qwn, which is essentially $q\log(q)$ since q is exponential in n. Constructing explicit permutations with low conductance and low all-but-one conductance remains a nice open problem.

5 Network Nomenclature and Main Result

In this section we start with a syntax-oriented description of the confusion-diffusion networks for which our results are obtained. We follow with the formal statement of our main theorem.

While we mentioned six different networks in the introduction (cf. Fig. 1), our full results actually encompass two extra networks, of 6 and 10 rounds (but structurally different from the "other" 6 and 10 round networks); these two extra networks are not mentioned in Fig. 1 because they offer no structural, security or query complexity advantages over other networks. However we include them what follows since they come anyway "for free", so that our result will subsume eight different networks. (Moreover the proof is easier to write this way, since the eight different networks happen to arise as the result of setting three independent boolean flags).

Network Nomenclature. A *round* of a confusion-diffusion network refers to a round of S-boxes. More precisely, all S-box permutations $P_{i,j}$ with the same value of i lie in the same *round* of the network.

Since, say, the middle round of our 5-round confusion-diffusion network plays the same structural role (with respect to our simulator) as the middle round in our 9-round network, it makes more sense to designate rounds according to their

structural role instead of by their round number (as the latter will keep changing from network to network, even while the structural purpose of the round stays the same).

For this purpose, we replace the array $\mathcal{P} = \{P_{i,j}\}$ of $r \times w$ random permutations with an array $\mathcal{Q}$ of $12 \times w$ random permutations where each "round" (value of the index i) is designated by a different alphabet letter. Specifically, we let

$$\mathcal{Q} = \{F_j, G_j, I_j, D_j, J_j, B_j, A_j, C_j, K_j, E_j, L_j, H_j : j \in [w]\} \tag{1}$$

be a set of 12 w random permutations, where each permutation is thus indexed by an alphabet letter from the set $\{A, \ldots, L\}$ as well as by an index $j \in [w]$.

Having traded the set of indices $\{i : i \in [r]\}$ (the possible round numbers) for the set of letters $\{A, \ldots, L\}$, a "round" will henceforth mean a member of the latter set, i.e., a "round" means one of the letters $A, \ldots, L$.

Not all rounds will be used for all confusion-diffusion networks, and we describe which rounds appear in which networks below. (See also Fig. 2 below as an aid). However two rounds always appear in the same order as they appear listed in $\mathcal{Q}$ above (cf. (1)), when they both appear in a network.

In more detail, our eight different confusion-diffusion networks correspond to the eight different possible settings of three independent boolean flags called XtraMiddleRnd, XtraOuterRnd and XtraUntglRnds. The rounds that appear in each network, as a function of these boolean flags, are as follows:

$$\begin{cases} A & \text{if XtraMiddleRnd is off} \\ B, C & \text{if XtraMiddleRnd is on} \end{cases}$$

$$\begin{cases} G, H & \text{if XtraOuterRnd is off} \\ F, G, H & \text{if XtraOuterRnd is on} \end{cases}$$

$$\begin{cases} D, E & \text{if XtraUntglRnds is off} \\ I, D, J, K, E, L & \text{if XtraUntglRnds is on} \end{cases}$$

As can be seen, toggling either of XtraMiddleRnd or XtraOuterRnd "costs" one extra round, whereas toggling XtraUntglRnds costs four extra rounds. Hence, and because the network has 5 rounds when all flags are off, the number of rounds in the network will be

$$5 + \mathsf{XtraMiddleRnd} + \mathsf{XtraOuterRnd} + 4 \cdot \mathsf{XtraUntglRnds}$$

which spans the integers 5, 6, 6, 7, 9, 10, 10, 11. In Fig. 1 the flag bits appear in the order XtraMiddleRnd, XtraOuterRnd, XtraUntglRnds, hence the two "missing" combinations are those where XtraMiddleRnd/XtraOuterRnd are off/on. In addition, the top half of the table is missing all rows with XtraUntglRnds = **true** since toggling this flag does not (significantly) improve security or query complexity when the D-boxes can be arbitrary and, in particular, when the D-boxes can be RCR.

For example, our 11-round network consists of the rounds

$$F, G, I, D, J, B, C, K, E, L, H$$

in this order. (We refer to Fig. 2 further down). The 10-round network with XtraMiddleRnd = **false** consists of the rounds

$$F, G, I, D, J, A, K, E, L, H$$

in this order as well. All other networks can be obtained by removing rounds from one of these two sequences. In more detail, round F is removed to un-toggle XtraOuterRnd and rounds I, J, K, L are removed to un-toggle XtraUntglRnds.

We will also rename the generic diffusion permutations $\overline{\pi} = (\pi_1, \ldots, \pi_r)$ according to their structural roles in the diffusion network. Specifically, we let

$$\overline{\pi} = (\nu, \pi_G, \pi_I, \pi_J, \pi_B, \tau, \pi_C, \pi_K, \pi_L, \pi_H)$$

where each element in the sequence $\overline{\pi}$ is a permutation from $\{0,1\}^{wn}$ to $\{0,1\}^{wn}$. (Thus, we redefine $\overline{\pi}$ to a specific sequence of diffusion permutations). In the 11-round confusion-diffusion network, diffusion permutations appear interleaved with the S-box rounds in the order

$$F\text{–}\nu\text{–}G\text{–}\pi_G\text{–}I\text{–}\pi_I\text{–}D\text{–}\pi_J\text{–}J\text{–}\pi_B\text{–}B\text{–}\tau\text{–}C\text{–}\pi_C\text{–}K\text{–}\pi_K\text{–}E\text{–}\pi_L\text{–}L\text{–}\pi_H\text{–}H$$

(i.e., the S-box round consisting of the parallel application of the permutations F_j is followed by the diffusion permutation ν, and so on), whereas in the 10-round network with XtraMiddleRnd= **false** the diffusion permutations appear in the order

$$F\text{–}\nu\text{–}G\text{–}\pi_G\text{–}I\text{–}\pi_I\text{–}D\text{–}\pi_J\text{–}J\text{–}\pi_B\text{–}A\text{–}\pi_C\text{–}K\text{–}\pi_K\text{–}E\text{–}\pi_L\text{–}L\text{–}\pi_H\text{–}H$$

with τ dropped. From either of these configurations one can un-toggle XtraOuterRnd by dropping F–ν– and one can un-toggle XtraUntglRnds by dropping I–π_I–, –π_J–J, K–π_K– and –π_L–L. For example, our 9-round confusion-diffusion network has the order

$$G\text{–}\pi_G\text{–}I\text{–}\pi_I\text{–}D\text{–}\pi_J\text{–}J\text{–}\pi_B\text{–}A\text{–}\pi_C\text{–}K\text{–}\pi_K\text{–}E\text{–}\pi_L\text{–}L\text{–}\pi_H\text{–}H$$

whereas the 5-round and 6-round network with XtraMiddleRnd toggled respectively have order

$$G\text{–}\pi_G\text{–}D\text{–}\pi_B\text{–}A\text{–}\pi_C\text{–}E\text{–}\pi_H\text{–}H$$
$$G\text{–}\pi_G\text{–}D\text{–}\pi_B\text{–}B\text{–}\tau\text{–}C\text{–}\pi_C\text{–}E\text{–}\pi_H\text{–}H$$

and so on.

In summary, the confusion-diffusion network under consideration is a function of the confusion permutations $\mathcal{Q}$, of the diffusion permutation vector $\overline{\pi}$ and of

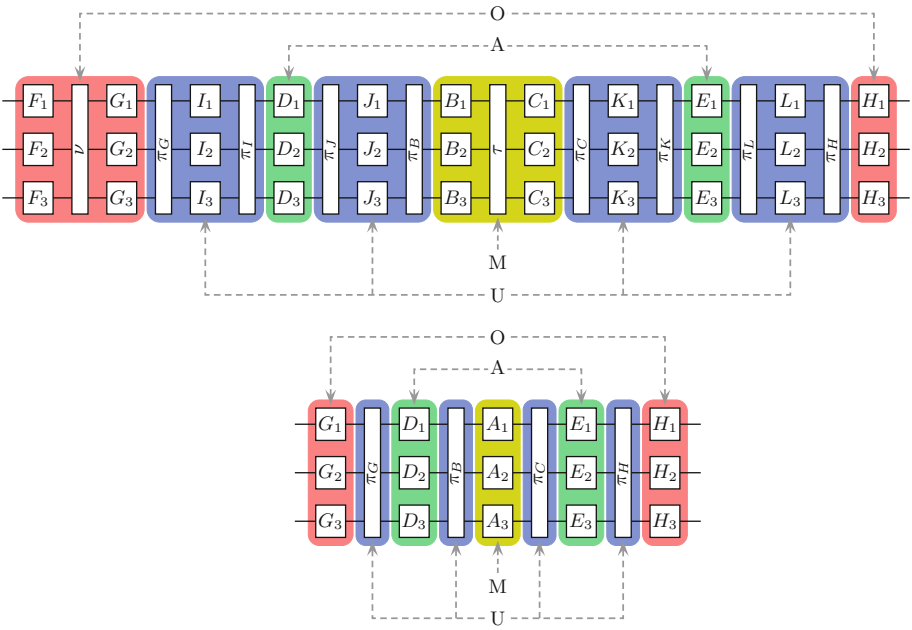

Fig. 2. Emplacement of the outer detect (O), adapt (A), middle detect (M) and untangle (U) zones for the 11- and 5-round simulators. The adapt zones always consist of rounds D and E.

the three boolean flags XtraMiddleRnd, XtraOuterRnd and XtraUntglRnds. For brevity we write this network as

$$P[\mathcal{Q}, \overline{\pi}]$$

keeping the three boolean flags implicit. Depending on the value of the flags some permutations in $\mathcal{Q}$ and/or $\overline{\pi}$ are of course unused. In particular, we assume that unused permutations in $\mathcal{Q}$ are simply ignored for the purpose of the indifferentiability experiment (i.e., these unused permutations are not accessible as oracles).

Semantically, moreover, our simulator divides the confusion rounds and diffusion permutations into nine "zones" of four different types, to wit, one *middle detect* zone (M), left and right *outer detect* zones (O), four *untangle* zones (U) and two *adapt* zones (A). Each zone consists of one or more contiguous rounds and/or diffusion permutations, with every round and every diffusion permutation belonging to exactly one zone. Figure 2 shows how the zones appear in the 11- and 5-round networks, while the zoning of other networks can be interpolated from these, given that each of our zones either appears the same as in the 11-round network or the same as in the 5-round network. (For example, if XtraMiddleRnd is off, then round A exists and rounds B and C do not, so the middle detect zone consists of round A only, as in the 5-round simulator; etc).

In particular,

XtraMiddleRnd

determines the aspect of the middle detect zone (and nothing else), while

XtraOuterRnd

determines the aspect of the left outer detect zone (and nothing else), and

XtraUntglRnds

determines the aspect of the untangle zones (and nothing else).

In particular, the D-box τ appears in the middle detect zone if it appears at all, the D-box ν appears in the left outer detect zone if it appears at all, and remaining D-boxes (of the form $\pi_{\ldots}$) appear in the untangle zones. Further simulator details, including the purpose of the zones, are given in Sect. 6.

Main Result: Takeaway Points. Before giving our full technical result (which can be difficult to parse the first time around), we summarize its key implications for security, query complexity, and D-box properties at a high level:

Security. The security of our simulator is essentially a function of the middle detect zone and of the conductance of τ. If XtraMiddleRnd is off, more precisely, so that the middle detect zone consists only of round A, then security is approximately q^{2w}/N where $N = 2^n$, whereas if XtraMiddleRnd is on, the security improves (essentially) to q^2/N, assuming that τ has conductance $\sim q$.

Query Complexity. The simulator's query complexity is determined by the left outer detect zone and by the middle detect zone. The query complexity is approximately q^{w^2} if neither of XtraMiddleRnd or XtraOuterRnd is toggled, is approximately q^w if exactly one of these two flags is toggled, and is approximately q if both flags are toggled, where the bounds quoted in the last two cases assume that τ and/or ν both have conductance $\sim q$.

Combinatorial Requirements on the Diffusion Permutations. The diffusion permutations τ and ν must have low conductance and low all-but-one-conductance. (Near q in order to have bounds approaching those of Fig. 1). The combinatorial requirements on the permutations $\pi_{\ldots}$ are different depending on whether XtraUntglRnds is toggled or not:

- if XtraUntglRnds is off the permutations in the untangle zones must be both RPR- and RCR-resistant (or else the above-quoted security bounds are not achieved); in particular, if $w > 2$ then these permutations *cannot* be $\mathrm{GF}(2^n)$-linear, because, as mentioned above, $\mathrm{GF}(2^n)$-linear permutations are not RCR-resistant for $w > 2$;
- if XtraUntglRnds is on then the permutations in the untangle zones need only be RPR-resistant; in particular, these permutations can now be $\mathrm{GF}(2^n)$-linear even for $w > 2$.

One can also summarize the situation orthogonally, according to which boolean flags have which effect. Namely:

- toggling XtraMiddleRnd impacts security and query complexity
- toggling XtraOuterRnd impacts query complexity
- toggling XtraUntglRnds reduces the cryptographic requirements on the permutations $\pi_{\ldots}$

It is natural to wonder whether even further extension of the middle, outer left, or untangle zones can be beneficial. The short answer to this question, that we consider in detail in [8], is yes; namely, by adding more rounds to these zones one can further reduce the combinatorial requirements on the diffusion permutations, leading, e.g., to networks in which the permutations in the untangle zones are not even RPR-resistant.

As a final high-level comment, a potentially striking feature of our simulator is the left-right asymmetry that arises when XtraOuterRnd is toggled. As explained in Sect. 6, adding an extra round to the right-hand outer detect zone (assuming XtraOuterRnd is already set) does not further decrease the simulator's query complexity, nor improve security. Nonetheless, such an extra zone can be used to reduce the simulator's *space complexity* while maintaining essentially the same security and query complexity. (See the discussion after Lemma 52 in the full version). Hence, while the benefit of adding an additional round to the right outer detect zone is somewhat technical, such an extra round does have a potential justification in terms of simulator design.

Main Result. We recall that $\overline{\pi} = (\nu, \pi_G, \pi_I, \pi_J, \pi_B, \tau, \pi_C, \pi_K, \pi_L, \pi_H)$ is the vector of diffusion permutations, not necessarily all of which are used. In order to more succinctly state the main result, we define

$$\begin{aligned}
\mathsf{MaxColl}(\overline{\pi}) &= \max(\mathsf{MaxColl}(\pi_G), \mathsf{MaxColl}(\pi_B^{-1}), \mathsf{MaxColl}(\pi_C), \mathsf{MaxColl}(\pi_H^{-1})) \\
\mathsf{MaxPreim}(\overline{\pi}) &= \max(\mathsf{MaxPreim}(\pi_G), \mathsf{MaxPreim}(\pi_B^{-1}), \mathsf{MaxPreim}(\pi_C), \mathsf{MaxPreim}(\pi_H^{-1}), \\
&\qquad \mathsf{MaxPreim}(\pi_I), \mathsf{MaxPreim}(\pi_J^{-1}), \mathsf{MaxPreim}(\pi_K), \mathsf{MaxPreim}(\pi_L^{-1})) \\
\mathsf{MaxCoPr}(\overline{\pi}) &= \max(\mathsf{MaxColl}(\overline{\pi}), \mathsf{MaxPreim}(\overline{\pi}))
\end{aligned}$$

where π^{-1} denotes the inverse of π.

Moreover we define

$$\alpha(q) = \begin{cases} (2q)^w & \text{if XtraMiddleRnd is off,} \\ \mathsf{Cond}_\tau(2q) & \text{if XtraMiddleRnd is on,} \end{cases}$$

$$\beta(q) = \begin{cases} (q + \alpha(q))^w & \text{if XtraOuterRnd is off,} \\ \mathsf{Cond}_\nu(q + \alpha(q)) & \text{if XtraOuterRnd is on.} \end{cases}$$

The definitions of $\alpha(q)$ and $\beta(q)$ might seem annoyingly technical right now. In Sect. 6 we provide more digestible semantic explanations for $\alpha(q)$ and $\beta(q)$.

Theorem 1. *Let $N = 2^n$. The confusion-diffusion network $P[\mathcal{Q}, \overline{\pi}]$ achieves (t_S, q_S, ε)-indifferentiability from a random permutation $\mathcal{Z} : \{0,1\}^{wn} \to \{0,1\}^{wn}$*

for ε equal to

$$\begin{aligned}
&\frac{\beta(q)(q+\alpha(q))^w}{N^w-q-\alpha(q)}+\frac{1}{N^w}+\frac{4w(q+\alpha(q))^2}{N-q-\alpha(q)}\\
&\quad+\frac{4wq\,\mathsf{aboCond}_\tau(2q)}{N-2q} && \text{if } \mathsf{XtraMiddleRnd} \text{ is on}\\
&\quad+\frac{2w(q+\alpha(q))\,\mathsf{aboCond}_\nu(q+\alpha(q))}{N-q-\alpha(q)} && \text{if } \mathsf{XtraOuterRnd} \text{ is on}\\
&\quad+\frac{4w\alpha(q)(q+\alpha(q))\mathsf{MaxCoPr}(\overline{\pi})}{N-q-\alpha(q)} && \text{if } \mathsf{XtraUntglRnds} \text{ is off}\\
&\quad+\frac{6w(q+\alpha(q))^2\,\mathsf{MaxPreim}(\overline{\pi})}{N-q-\alpha(q)} && \text{if } \mathsf{XtraUntglRnds} \text{ is on}
\end{aligned}$$

and for $q_S=\beta(q)$, $t_S=O(w(q+\alpha(q))^w)$. Here $q=q_0(1+rw)$ where q_0 is the number of distinguisher queries and $r\in\{5,6,7,9,10,11\}$ is the number of rounds in the confusion-diffusion network.

The proof of this result is in the full version [8].

Parsing the Security Bound. In order to get a rough feel for the security bound of Theorem 1 it is helpful to make the order-of-magnitude approximations

$$\begin{aligned}
\mathsf{MaxPreim}(\overline{\pi})=\mathsf{MaxColl}(\overline{\pi})&\approx O(1)\\
\mathsf{Cond}_\tau(2q)=\mathsf{aboCond}_\tau(2q)&\approx q\\
\mathsf{Cond}_\nu(q+\alpha(q))=\mathsf{aboCond}_\nu(q+\alpha(q))&\approx \alpha(q).
\end{aligned}$$

With these approximations in place, and given $q\ll N$ (in fact we can assume $q\le N^{1/2}$, since the security bound is void otherwise) it easy to verify that the largest terms in Theorem 1 are of the order

$$\frac{\alpha(q)^2}{N}$$

and which is, therefore, a first approximation to the security ε that we achieve. Since

$$\alpha(q)=\begin{cases}(2q)^w & \text{if } \mathsf{XtraMiddleRnd} \text{ is off,}\\ q & \text{if } \mathsf{XtraMiddleRnd} \text{ is on,}\end{cases}$$

under the order-of-magnitude approximations given above, we find

$$\varepsilon\approx\begin{cases}(2q)^{2w}/N & \text{if } \mathsf{XtraMiddleRnd} \text{ is off,}\\ q^2/N & \text{if } \mathsf{XtraMiddleRnd} \text{ is on}\end{cases}$$

for the security, to a first approximation. On the other hand we find

$$q_S=\beta(q)\approx\begin{cases}(2q)^{w^2} & \text{if } \mathsf{XtraMiddleRnd}/\mathsf{XtraOuterRnd} \text{ are off/off}\\ (2q)^w & \text{if } \mathsf{XtraMiddleRnd}/\mathsf{XtraOuterRnd} \text{ are off/on}\\ (2q)^w & \text{if } \mathsf{XtraMiddleRnd}/\mathsf{XtraOuterRnd} \text{ are on/off}\\ q & \text{if } \mathsf{XtraMiddleRnd}/\mathsf{XtraOuterRnd} \text{ are on/on}\end{cases}$$

for the query complexity, again to a first approximation.

On the other hand, it is relatively easy to see that $\mathsf{MaxColl}(\pi) = 2^n = N$ for any linear permutation $\pi : \mathrm{GF}(2^n)^w \to \mathrm{GF}(2^n)^w$ as long as $w > 2$. In this case $\mathsf{MaxCoPr}(\pi) = N$, and Theorem 1 becomes void if $\mathsf{XtraUntglRnds}$ is off. Thus one of the main reasons for toggling $\mathsf{XtraUntglRnds}$ might be to enable the use of linear diffusion permutations, or any other (fast) family of permutations that have small entry-wise randomized preimage resistance (but potentially large entry-wise randomized collision resistance).

6 Simulator Overview

Context. We start with some very high-level description and reminder-of-purpose of our simulator. For this discussion it will be more convenient if we momentarily revert to indexing the S-boxes by coordinate pairs (i, j) where $i \in [r]$ the round number and $j \in [w]$ the layer number, with r being the number of rounds and w being the width. The diffusion permutation between the i-th and $(i + 1)$-th rounds will again be denoted π_i as well.

The simulator is responsible for answering queries to the S-boxes, and has access to a random permutation oracle $\mathcal{Z} : \{0,1\}^{wn} \to \{0,1\}^{wn}$ that is being independently accessed by the distinguisher. The simulator's job is to keep the S-box answers compatible with $\mathcal{Z}$ in the sense that it looks to the distinguisher as if $\mathcal{Z}$ is implemented by the confusion-diffusion network.

For each pair $(i, j) \in [r] \times [w]$ the simulator maintains a pair of tables $P_{i,j}$ and $P_{i,j}^{-1}$, each containing 2^n entries of n bits each, in which the simulator keeps a record of "what it has already decided" about the (i, j)-th S-box. Initially the tables are blank, meaning that $P_{i,j}(x) = P_{i,j}^{-1}(y) = \perp$ for all $x, y \in \{0,1\}^n$. The simulator sets $P_{i,j}(x) = y$, $P_{i,j}^{-1}(y) = x$ to indicate that the (i, j)-th S-box maps x to y. The simulator never overwrites values in $P_{i,j}$ or in $P_{i,j}^{-1}$ and always keeps these two tables consistent. Hence $P_{i,j}$ encodes a partial matching (or "partial permutation") from $\{0,1\}^n$ to $\{0,1\}^n$ from which edges are never subtracted. We also note that the edges in $P_{i,j}$ are a *superset* of those queries that the distinguisher has made to the (i, j)-th S-box or to its inverse (i.e., $P_{i,j}$ contains the answers to those queries, and possibly more).

By analogy with the notation of Sect. 2 we write

$$P_i(\mathbf{x}) = \mathbf{y} \tag{2}$$

if $\mathbf{x}, \mathbf{y} \in \{0,1\}^{wn}$ are vectors such that $P_{i,j}(\mathbf{x}[j]) = \mathbf{y}[j]$ for all $j \in [w]$. Note that (2) is a time-dependent statement, in the sense that the tables $P_{i,j}$ keep accruing entries as the distinguishing experiment proceeds. For example, (2) is initially false for all i and all vectors $\mathbf{x}$, $\mathbf{y}$. Moreover P_i is not an actual table maintained by the simulator—i.e., (2) is "just notation".

A sequence of vectors $(\mathbf{x}^1, \mathbf{y}^1, \ldots, \mathbf{x}^r, \mathbf{y}^r)$ is called a *completed path*[6] if $P_i(\mathbf{x}^i) = \mathbf{y}^i$ for $i = 1, \ldots, r$ and if $\pi_i(\mathbf{y}^i) = \mathbf{x}^{i+1}$ for $i = 1, \ldots, r-1$. The set of completed paths is also time-dependent. The vectors $\mathbf{x}^1$ and $\mathbf{y}^r$ are called the *endpoints* of the path.

We informally say that the distinguisher *completes a path* if it makes queries to the simulator that form a completed path. (There are many different possible ways to order such a set of queries, obviously). One can picture the distinguisher as trying to complete paths in various devious ways (typically, reusing the same queries as part of different paths), and checking that the path endpoints are each time compatible with $\mathcal{Z}$.

The simulator's job, in response, is to run ahead of the distinguisher and pre-emptively complete paths that it thinks the distinguisher is interested in, such as to make these paths are compatible with $\mathcal{Z}$. The simulator's dilemna is that it must choose under which conditions to complete a path; if it waits too long, or completes paths in only highly specialized cases, it may find itself trapped in a contradiction (typically, while trying to complete several paths at once); but if it is too trigger-happy, having a very large number of conditions under which it will choose to complete a path, the simulator runs the risk creating[7] an out-of-control chain reaction of path completions.

Essentially the simulator must be safe, but in a smart enough way that it avoids (out-of-control) chain reactions. We will informally refer to the problem of showing that no out-of-control chain reactions occur as the problem of *simulator termination*.

Simulator Zones. As already mentioned our simulator divides the confusion rounds and diffusion permutations into nine zones of four different types, shown in Fig. 2 for the 11- and 5-round networks.

Specifically, the "middle zone" is

$$\begin{cases} A & \text{if XtraMiddleRnd is off} \\ B\text{–}\tau\text{–}C & \text{if XtraMiddleRnd is on} \end{cases}$$

depending only on XtraMiddleRnd; the left and right outer detect zones are

$$\begin{cases} G, H & \text{if XtraOuterRnd is off} \\ F\text{–}\nu\text{–}G, H & \text{if XtraOuterRnd is on} \end{cases}$$

[6] This definition, made for the sake of expository convenience, is superceded further down, where we redefine "completed path" by adding the requirement that the endpoints be compatible with $\mathcal{Z}$, i.e., that $\mathcal{Z}(\mathbf{x}^1) = \mathbf{y}^r$.

[7] Indeed, the simulator makes no distinction between those entries in its tables $P_{i,j}$ that are the direct result of an distinguisher query, and those which it created on its own while pre-emptively completing paths. It seems very hard to leverage such a distinction. Note for example that the distinguisher may know values in $P_{i,j}$ without having made the relevant queries, simply by virtue of knowing how the simulator works.

depending only on XtraOuterRnd; the four "untangle zones" are

$$\begin{cases} \pi_G, \pi_B, \pi_C, \pi_H & \text{if XtraUntglRnds is off} \\ \pi_G\text{–}I\text{–}\pi_I, \pi_J\text{–}J\text{–}\pi_B, \pi_C\text{–}K\text{–}\pi_K, \pi_L\text{–}L\text{–}\pi_H & \text{if XtraUntglRnds is on} \end{cases}$$

depending only on XtraUntglRnds; the two "adapt" zones, finally, are rounds D and E in all networks.

We observe that zones (specifically, untangle zones with XtraUntglRnds = **false**) might consist solely of diffusion permutations. This contrasts with semantically similar zoning systems for Feistel network simulators and key-alternating simulators [18,19,27] for which the zones always contain at least one ideal component. We also observe that in the minimalist 5-round simulator, each round and each diffusion permutation corresponds to an individual zone.

Table Notation, Columns and Matching Query Sets. We revert to identifying rounds with letters in $\{A, \ldots, L\}$. Under this notation tables $P_{i,j}$, $P_{i,j}^{-1}$ described above become, e.g., tables A_j and A_j^{-1}. Thus for each round $T \in \{A, \ldots, L\}$ that is "operational" (as will depend on the flag settings) the simulator maintains tables T_j, T_j^{-1} for $1 \leq j \leq w$, as already described above under the notation $P_{i,j}, P_{i,j}^{-1}$.

We write $T(\mathbf{x}) = \mathbf{y}$ if $T_j(\mathbf{x}[j]) = \mathbf{y}[j]$ for each $j \in [w]$, and for all $T \in \{A, \ldots, L\}$. We also write $T(\mathbf{x}) = \perp$ if $T_j(\mathbf{x}[j]) = \perp$ for at least one value of j. The notation $T^{-1}(\mathbf{y})$ is analogous, with T^{-1} being the inverse of T. As emphasized above this notation is really "notation only" in the sense that the simulator does not maintain such things as tables T or T^{-1}.

A non-null entry in table T_j will be called an (S-box) *query*. More formally, an S-box query is a quadruple (T, j, x, y) where $T \in \{A, \ldots, L\}$, $j \in [w]$, $x, y \in \{0,1\}^n$, such that $T_j(x) = y$.

A set of w queries with the same value of T but with different values of j will be called a *column* or a *T-column* when we wish to emphasize the round. The (unique) vectors $\mathbf{x}, \mathbf{y}$ such that $(T, j, \mathbf{x}[j], \mathbf{y}[j])$ is a query in the column for each $j \in [w]$ are called the *input* and *output* of the column respectively. We note that a column is uniquely determined by either its input or output.

Two columns are *adjacent* if their rounds are adjacent. (E.g., a B-column and a C-column are adjacent). Two adjacent columns are *matching* if $\pi(\mathbf{y}) = \mathbf{x}$, where $\mathbf{y}$ is the output of the first column, where $\mathbf{x}$ the input of the second column, and where π is the diffusion permutation between the two rounds.

A pair of columns from the first and last round of the confusion-diffusion network are likewise *matching* if $\mathcal{Z}(\mathbf{x}) = \mathbf{y}$, where $\mathbf{x}$ is the input to the first-round column (either an F- or G-column) and $\mathbf{y}$ is the output of the last-round column (the H-column).

The notion of matching columns naturally extends to sequences of columns from consecutive rounds of length greater than two. (The first and last round of the network are always considered adjacent). If a set of matching columns encompasses all rounds we call the set a *completed path*. Thus, since the first and last column of a network are considered adjacent, completed paths are compatible with $\mathcal{Z}$ by definition.

The set of queries in a matching sequence of columns of any length is called a *matching set* of queries. We will also refer to the queries of a single column as a matching set, considering that column as a matching sequence of columns of length 1.

One can also observe that two different completed paths cannot contain the same column. Indeed, each D-box is a permutation and each round implements a partial permutation as well.

Simulator Operation and Termination Argument. Our simulator applies a design paradigm pioneered by Seurin [27] that has also been used in other simulators for Feistel networks or key-alternating ciphers [18,19]. As for all such simulators, our simulator completes two types of paths, where one type of path is triggered by a "middle detect zone" and the other is triggered by an "outer detect zone".

In more detail, our simulator pre-emptively completes a path[8] for every matching set of queries in the middle detect zone (such a matching set will consist of either one or two columns, depending on whether XtraMiddleRnd is toggled) and also for every matching set of queries in the two outer detect zones, considered as a single consecutive set of columns (the latter kind of matching set will thus consist of either two or three columns, depending on whether XtraOuterRnd is toggled).

Crucially, one can show that, unless some bad event of negligible probability occurs, each outer-triggered path completion is associated to a unique pre-existing query made by the distinguisher to $\mathcal{Z}$. Since the distinguisher makes only q queries in total, this means that at most q outer-triggered path completions occur, with high probability. In fact our simulator aborts if it counts[9] more than q outer-triggered path completions, so in our case at most q outer-triggered path completions occur with probability 1.

Moreover, a middle-triggered path completion does not add any new queries to the middle detect zone. This means that all queries in the middle detect zone can be chalked up to one of two causes: (1) queries directly made by the distinguisher, (2) queries made by the simulator during outer-triggered path completions.

Cause (1) accounts for at most q queries to each S-box and, as just seen, cause (2) accounts for at most q queries as well. Hence a middle detect zone S-box is never queried at more then $2q$ points, i.e., the table T_j for $T \in \{A, B, C\}$ never has more than $2q$ non-$\perp$ entries.

[8] The phrase "completes a path" is informal at this point, as there are generally many different ways to complete a path. (E.g., where to "adapt" a path to make it compatible with $\mathcal{Z}$, etc). More details follow below.

[9] This means the simulator knows the value of q beforehand, which introduces a small amount of non-uniformity into the simulator. One could remove this non-uniformity—i.e., not tell the simulator the value of q beforehand—at the cost of a more complicated theorem statement and proof. But the fact that essentially all security games allow adversaries that know the value of q for which they want to carry out an attack makes the issue a bit moot.

In particular, if XtraMiddleRnd is off, the latter implies that no more than $(2q)^w$ middle-triggered path completions occur, or one for every possible combination of an entry from each of the tables $A_1, \ldots, A_w$. If XtraMiddleRnd is on, on the other hand, than at most $\mathsf{Cond}_\tau(2q)$ middle-triggered path completions occur, as is easy to see per the definition[10] of conductance. In other words, $\alpha(q)$ (cf. Sect. 5) is an upper bound on the number of middle-triggered path completions. In fact, $\alpha(q)$ is an upper bound for the *total* number of path completions performed (including also outer path completions), since each completed path is also associated to a unique set of matching queries from the middle detect zone.

As for all S-boxes outside the middle detect zone, their queries can also be chalked up to one of two sources, namely direct distinguisher queries and path completions. There are at most q direct distinguisher queries, and each completed path contributes at most 1 query to each S-box, so each S-box outside the middle detect zone ends up with at most $q + \alpha(q)$ queries.

The simulator, moreover, only queries $\mathcal{Z}$ in order to either complete paths or else in order to detect outer-triggered path completions. This implies the number of distinct simulator queries to $\mathcal{Z}$ is upper bounded by the number of matching sets of queries in the left-hand outer detect zone. Indeed each completed path is obviously associated to a matching set of queries in the left-hand outer detect zone; and for the purpose of outer-triggered path detection, it is easy to see that the simulator only needs to query $\mathcal{Z}$ at most once for each such matching set as well by maintaining a table[11] of queries already made to $\mathcal{Z}$. If XtraOuterRnd is off, thus, the number of simulator queries to $\mathcal{Z}$ will be at most $(q + \alpha(q))^w$; whereas if XtraOuterRnd is on, the same number will be at most $\mathsf{Cond}_\nu(q+\alpha(q))$. The simulator query complexity is thus at most $\beta(q)$ (cf. Sect. 5).

7 Extensions

In [8] we discuss the extension of our main result to even longer networks. In particular, we show that each of our four combinatorial properties are specialized (extreme) cases of more general adversarial games involving entire segments of CD networks. We give a more general theorem (albeit without proof) in terms of these properties defined on network segments. The use of more rounds, then, enables the D-boxes to become even weaker (provably so in the case of the untangle rounds, conjecturally so for the detect zones), and thus even faster. This leaves many exciting possibilities/questions for future work.

[10] More precisely, let $U_j = \{y \in \{0,1\}^n : B_j^{-1}(y) \neq \perp$, let $V_j = \{x \in \{0,1\}^n : C_j(x) \neq \perp\}$ at the end of the distinguishing experiment. Then $|U_j|, |V_j| \leq 2q$ for each j, and the number of middle-triggered path completions that have occurred is at most

$$\{(\mathbf{x}, \mathbf{y}) : \tau(\mathbf{x}) = \mathbf{y}, \mathbf{x} \in \prod_j U_j, \mathbf{y} \in \prod_j V_j\} \leq \mathsf{Cond}_\tau(2q).$$

[11] Thus, in particular, adding an extra round to the right-hand detect zone would not further reduce the query complexity, since the query complexity is only proportional to the number of matching query sets for the left-hand outer detect zone.

Another possible improvement is to reduce the number of independent S-boxes. In a private note (unpublished), we have proved an analogous result where the same S-box is used at each round. To make this variant indifferentiable from a random permutation over $\{0,1\}^{wn}$, the D-boxes need to satisfy stronger combinatorial properties. For example, the current construction needs a D-box with entry-wise randomized preimage resistance (RPR): roughly, if one of the input blocks is uniform random and the other input blocks are fixed, then each output block is fairly random. In the variant construction with identical S-boxes, the D-boxes must satisfy a generalization of RPR: roughly, if some of the input blocks contain the same uniform random value and the other input blocks are fixed, then each output block is fairly random. We also found that a minor modification to our D-boxes would satisfy these generalized properties. Additionally, in this variant construction the security (distinguisher's advantage) ε and simulator query complexity q_S are worse than the current construction. The query complexity is increased by a factor of w^{w^2}, w^w or w (depending on the flags), while the security deteriorates by a factor of w.

References

1. Andreeva, E., Bogdanov, A., Dodis, Y., Mennink, B., Steinberger, J.P.: On the indifferentiability of key-alternating ciphers. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 531–550. Springer, Heidelberg (2013)
2. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: On the indifferentiability of the sponge construction. In: Smart [31] pp. 181–197
3. Biryukov, A., Bouillaguet, C., Khovratovich, D.: Cryptographic schemes based on the ASASA structure: black-box, white-box, and public-key (Extended Abstract). In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014. LNCS, vol. 8873, pp. 63–84. Springer, Heidelberg (2014)
4. Chakraborty, D., Sarkar, P.: A new mode of encryption providing a tweakable strong pseudo-random permutation. In: Robshaw, M. (ed.) FSE 2006. LNCS, vol. 4047, pp. 293–309. Springer, Heidelberg (2006)
5. Chakraborty, D., Sarkar, P.: HCH: a new tweakable enciphering scheme using the hash-encrypt-hash approach. In: Barua, R., Lange, T. (eds.) INDOCRYPT 2006. LNCS, vol. 4329, pp. 287–302. Springer, Heidelberg (2006)
6. Chow, S., Eisen, P.A., Johnson, H., van Oorschot, P.C.: White-box cryptography and an AES implementation. In: Nyberg, K., Heys, H.M. (eds.) SAC 2002. LNCS, vol. 2595, pp. 250–270. Springer, Heidelberg (2003)
7. Chow, S., Eisen, P., Johnson, H., van Oorschot, P.C.: A white-box DES implementation for DRM applications. In: Feigenbaum, J. (ed.) DRM 2002. LNCS, vol. 2696, pp. 1–15. Springer, Heidelberg (2003)
8. Dodis, Y., Tianren, L., Stam, M., Steinberger, J.: Indifferentiability of Confusion-Diffusion Networks, IACR eprint archive 2015/680. (Full version of this paper.)
9. 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)
10. 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)

11. Coron, J.-S., Patarin, J., Seurin, Y.: The random oracle model and the ideal cipher model are equivalent. In: Wagner [32], pp. 1–20
12. Dodis, Y., Pietrzak, K., Puniya, P.: A new mode of operation for block ciphers and length-preserving macs. In: Smart [31], pp. 198–219
13. Dodis, Y., Reyzin, L., Rivest, R.L., Shen, E.: Indifferentiability of permutation-based compression functions and tree-based modes of operation, with applications to MD6. In: Dunkelman, O. (ed.) FSE 2009. LNCS, vol. 5665, pp. 104–121. Springer, Heidelberg (2009)
14. 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)
15. Feistel, H.: Cryptographic coding for data-bank privacy. IBM Technical report RC-2827, 18 March 1970
16. Fluhrer, S.R., McGrew, D.A.: The extended codebook (XCB) mode of operation. Technical report 2004/078, IACR eprint archive (2004)
17. Halevi, S.: Invertible universal hashing and the TET encryption mode. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 412–429. Springer, Heidelberg (2007)
18. Holenstein, T., Künzler, R., Tessaro, S.: The equivalence of the random oracle model and the ideal cipher model, revisited. In: Fortnow, L., Vadhan, S.P. (eds.), Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, pp. 89–98. ACM, 6–8 June 2011
19. Lampe, R., Seurin, Y.: How to construct an ideal cipher from a small set of public permutations. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part I. LNCS, vol. 8269, pp. 444–463. Springer, Heidelberg (2013)
20. Luby, M., Rackoff, C.: How to construct pseudorandom permutations and pseudorandom functions. SIAM J. Comput. **17**(2), 373–386 (1988)
21. Maurer, U.M., Renner, R.S., 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)
22. Maurer, U.M., Tessaro, S.: Domain extension of public random functions: beyond the birthday barrier. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 187–204. Springer, Heidelberg (2007)
23. Miles, E., Viola, E.: Substitution-permutation networks, pseudorandom functions, and natural proofs. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 68–85. Springer, Heidelberg (2012)
24. Naor, M., Reingold, O.: On the construction of pseudorandom permutations: Luby-Rackoff revisited. J. Cryptology **12**(1), 29–66 (1999). Preliminary Version: STOC
25. 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)
26. Rogaway, P., Steinberger, J.P.: Constructing cryptographic hash functions from fixed-key blockciphers. In: Wagner [32], pp. 433–450
27. Seurin, Y.: Primitives et protocoles cryptographiques à sécurité prouvée. Ph.D. thesis, Université de Versailles Saint-Quentin-en-Yvelines, France (2009)
28. Shannon, C.E.: Communication theory of secrecy systems. Bell Syst. Technical J. **28**(4), 656–715 (1949). www.cs.ucla.edu/jkong/research/security/shannon.html, www3.edgenet.net/dcowley/docs.html
29. Shrimpton, T., Stam, M.: Building a collision-resistant compression function from non-compressing primitives. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 643–654. Springer, Heidelberg (2008)

30. Smart, N.P. (ed.): EUROCRYPT 2008. LNCS, vol. 4965. Springer, Heidelberg (2008)
31. Boneh, D., Halevi, S., Hamburg, M., Ostrovsky, R.: Circular-secure encryption from decision diffie-hellman. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 108–125. Springer, Heidelberg (2008)
32. Wang, P., Feng, D., Wu, W.: HCTR: a variable-input-length enciphering mode. In: Feng, D., Lin, D., Yung, M. (eds.) CISC 2005. LNCS, vol. 3822, pp. 175–188. Springer, Heidelberg (2005)