Sequential Indifferentiability of Confusion-Diffusion Networks

Qi Da[1,2], Shanjie Xu[1,2], and Chun Guo[1,2,3](✉)

[1] Key Laboratory of Cryptologic Technology and Information Security of Ministry of Education, Shandong University, Qingdao, Shandong 266237, China
[2] School of Cyber Science and Technology, Shandong University, Qingdao, Shandong, China
daqi@mail.sdu.edu.cn, shanjie1997@gmail.com, chun.guo@sdu.edu.cn
[3] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China

Abstract. A large proportion of modern symmetric cryptographic building blocks are designed using the Substitution-Permutation Networks (SPNs), or more generally, Shannon's confusion-diffusion paradigm. To justify its theoretical soundness, Dodis et al. (EUROCRYPT 2016) recently introduced the theoretical model of *confusion-diffusion networks*, which may be viewed as keyless SPNs using *random permutations* as S-boxes and combinatorial primitives as permutation layers, and established provable security in the plain indifferentiability framework of Maurer, Renner, and Holenstein (TCC 2004).
We extend this work and consider Non-Linear Confusion-Diffusion Networks (NLCDNs), i.e., networks using *non-linear permutation layers*, in weaker indifferentiability settings. As the main result, we prove that 3-round NLCDNs achieve the notion of sequential indifferentiability of Mandal et al. (TCC 2012). We also exhibit an attack against 2-round NLCDNs, which shows the tightness of our positive result on 3 rounds. It implies correlation intractability of 3-round NLCDNs, a notion strongly related to known-key security of block ciphers and secure hash functions. Our results provide additional insights on understanding the complexity for known-key security, as well as using confusion-diffusion paradigm for designing cryptographic hash functions.

Keywords: Block ciphers · substitution-permutation networks · confusion-diffusion · indifferentiability · correlation intractability

1 Introduction

Modern block ciphers roughly fall into three classes. The first class consists of *Feistel networks and their generalizations*, with DES, LBlock [41], and many other block cipher standards as popular instances. The second class are the Lai-Massey structures designed for IDEA [28, 27]. This paper focuses on the last class, namely the *Substitution-Permutation Networks* (SPNs). Concretely, an SPN yields an wn-bit block cipher via iterating the following three steps:

1. *Key-addition*: XOR a round key with the wn-bit state;
2. *Substitution*: break down the wn-bit state into w disjoint chunks of n bits, and evaluate a small n-bit permutation, typically called an S-box, on each chunk;
3. *Permutation*: apply a keyless permutation to the whole wn-bit state.

The S-boxes are usually highly non-linear. On the other hand, while modern block ciphers tend to use linear or affine mappings for the *Permutation*, there is actually no a priori restriction, and the use and advantages of non-linear permutations was recently explored [29].

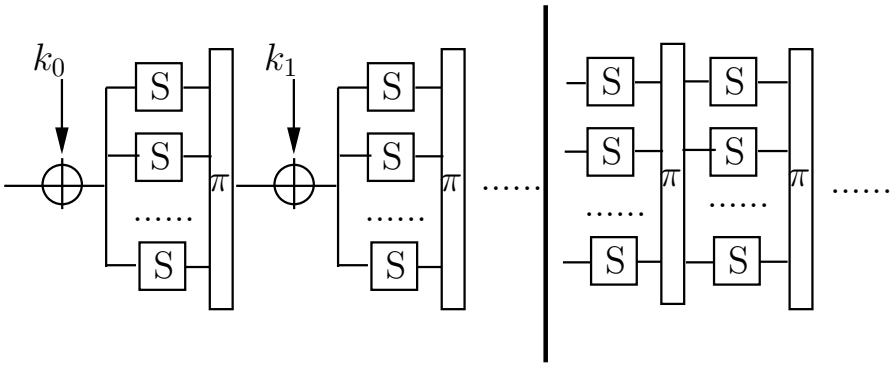

Fig. 1. Comparison of SPN and CDN, with SPN on the left and CDN on the right

The SPNs well fit into the *confusion-diffusion paradigm*: usually, the substitution is viewed as "confusion", while the permutation is viewed as "diffusion". The idea of confusion-diffusion goes back to the seminal paper of Feistel [17] and even back to Shannon [37]. Various popular primitives have been built upon this, including block ciphers such as the AES [13] and RECTANGLE [43] and hash functions such as the KECCAK-f permutations of the SHA [5]. Motivated by this popularity, SPNs have been the topic of various researches [12, 35, 40]. In particular, modeling the S-boxes as random or pseudorandom functions/permutation, SPNs can be proved as a strong pseudorandom permutation SPRP (i.e., indistinguishability from a truly random permutation), the standard security notion for block ciphers [25, 33, 7, 18].[1] We refer to [18] for a detailed survey of these SPRP results. In these security proofs, the S-boxes act as the only source of cryptographic hardness, while the permutation layers only supply auxiliary *combinatorial* properties. This limits the provable security to the domain-size of the S-boxes, which is unfortunately as small as 8 bits in, e.g., the AES. Consequently, provable results on SPNs do not relate to any concrete SPN-based block ciphers.

[1] We remark that, as proving such security for concrete block ciphers such as AES seems out of the reach of current techniques, it is actually the usual approach to idealize some underlying primitives and prove that the high-level structure meets certain security definitions.

Instead, they should be viewed as theoretical support for the SPN approach to constructing block ciphers. Indeed, the above results have confirmed (in a widely recognized theoretical model) that, the use of non-linear permutation layers ensures more security than linear ones. The provable bounds become meaningful when the "S-boxes" enjoy sufficiently large domains, e.g., when the "S-boxes" themselves are block ciphers such as the AES or cryptographic permutations such as the KECCAK-f. Therefore, on the practical side, the above results yield domain extension of block ciphers or permutations.

1.1 Indifferentiability of Confusion-Diffusion Networks

The aforementioned SPRP notion is formalized using the indistinguishability framework. A generalization of indistinguishability, named *indifferentiability*, was introduced by Maurer et al. [32]. Briefly, a construction $\mathcal{C}^{\mathcal{F}}$ built upon an ideal primitive $\mathcal{F}$ is indifferentiable from the ideal cryptographic primitive $\mathcal{G}$, if there exists an efficient simulator $\mathcal{S}^{\mathcal{G}}$ such that the two systems $(\mathcal{C}^{\mathcal{F}}, \mathcal{F})$ and $(\mathcal{G}, \mathcal{S}^{\mathcal{G}})$ are *indistinguishable*. The role of the simulator is to imitate the behavior of $\mathcal{F}$, such that it appears like the "underlying primitives" of the ideal primitive $\mathcal{G}$. The consistency of the simulation is possible by accessing $\mathcal{G}$.

Indifferentiability comes with a secure composition lemma, meaning that an indifferentiable cryptographic scheme could safely replace its ideal counterpart, even in the settings *with no secret keys*. Unsurprisingly, indifferentiability was soon adopted as a standard for evaluating cryptographic constructions, with applications to hash functions [10, 4], block cipher paradigms [11, 1, 22], and encryption schemes [3]. Due to this success, the authors won the TCC Test-of-Time award at TCC 2016-B [31].

The indifferentiability analysis of SPNs was initiated by Dodis et al. [16]. In detail, they introduced the model of *Confusion-Diffusion Networks* (CDNs), which may be viewed as SPNs *without key-additions*. In other words, CDNs is SPNs without key (see Fig. 1). Their CDN models are purely built upon public random S-boxes and non-cryptographic "D-boxes" (i.e., permutation layers), and indifferentiability measures the distance between such CDNs and wide random permutations. When the "D-boxes" are non-linear (and thus achieve a stronger diffusion), they showed that 5 rounds are sufficient for indifferentiability, and the concrete security bounds increase with the number of rounds. When the "D-boxes" are linear (as in common SPN ciphers), they showed that 9 rounds are sufficient for indifferentiability. This confirmed (in a widely recognized theoretical model) that, the use of non-linear diffusion layers ensures more security than linear ones. Dodis et al. also exhibited an attack against 2-round CDNs with arbitrarily strong (yet non-idealized) D-boxes [16, Section 3]. These justify the soundness of using fixed-key block ciphers as "random looking" permutations for constructing hash functions [36] and other sophisticated cryptosystems [21].

1.2 Weaker Variants of Indifferentiability

By incorporating different restrictions, the definition of indifferentiability has been generalized to various variants. Firstly, Yoneyama et al. [42], Dodis et al. [15], and Naito et al. [34] independently proposed the concept of *public indifferentiability*, in which the simulator $\mathcal{S}^{\mathcal{G}}$ is aware of all queries made by the distinguisher to the target ideal primitive $\mathcal{G}$. This captures the settings in which $\mathcal{G}$ only evaluated on public inputs, which fits into the use of, e.g., digital signatures. At TCC 2012, Mandal et al. [30] proposed another weakened variant named *sequential indifferentiability* (*seq-indifferentiability* for short), which restricts the distinguisher's queries to be "primitive-construction-sequential". Namely, the distinguisher consists of two phases. In the first phase, it queries the (simulated) "underlying primitive" $\mathcal{F}$ or $\mathcal{S}^{\mathcal{G}}$ in arbitrary, without making any query to the "construction" $\mathcal{C}^{\mathcal{F}}$ or $\mathcal{S}^{\mathcal{G}}$. In the second phase, it queries the "construction" $\mathcal{C}^{\mathcal{F}}$ or $\mathcal{G}$ in arbitrary, without making any query to the "primitive" $\mathcal{F}$ or $\mathcal{S}^{\mathcal{G}}$. It finally outputs the decision. Seq-indifferentiability is actually equivalent to the aforementioned public indifferentiability for natural constructions [30], while the former is easier to handle in the security analyses. In addition, seq-indifferentiability implies *correlation intractability* of Canetti et al. [6], i.e., there is no "non-trivial" relation between the inputs and outputs of the construction.

1.3 Our Results

As noted [39], indifferentiability appears imperfect for block cipher paradigms: security proofs are highly involved, and complexities of provably secure schemes appear far beyond necessary. In contrast, the notions of seq-indifferentiability and correlation intractability are directly linked to known-key security of block ciphers [26, 9], and are already sufficient for establishing security for block cipher-based hash functions. Due to these, several papers have characterized the seq-indifferentiability and correlation intractability of Feistel networks [30, 39] and variants of Even-Mansour ciphers [8, 23]. Though, the natural extension of this line of works to CD networks remains open.

With the above discussion, we characterize the sequential indifferentiability of NLCDNs, i.e., CD networks with non-linear D-boxes. [16] investigated full indifferentiability of CD networks (with both non-linear and linear D-boxes), while we study the weaker notion of sequential indifferentiability of CD networks with non-linear D-boxes only. As mentioned before, the motivation is that sequential indifferentiability was believed more suitable for known-key security of block ciphers, to some extent.

In this respect, our first observation is that Dodis et al.'s attack on 2-round NLCDNs [16, Section 3] is not sequential in any sense, and our first contribution is a primitive-construction-sequential distinguisher against 2-round NLCDNs with *any* (non-idealized) D-boxes. Depending on the D-boxes in use, the running time of our distinguisher may be exponential. Though, the *query complexity* is merely 2, indicating that 2-round CD networks are *insecure in the information theoretic setting*.

As positive results, we prove that 3-round NLCDNs are seq-indifferentiable, as long as the D-boxes satisfy some moderate conditions. The number of rounds is 40% less than that required for plain indifferentiability.[2] In addition, as discussed, the round complexity is *tight* in the information theoretic setting. As mentioned before, these imply that 3-round NLCDNs (tightly) achieve correlation intractability, and are thus sufficient for known-key security of CD networks (in the sense of correlation intractability).

Interpretations. Since initiated [26], models or adversarial goals for known-key attacks has incurred intensive discussion. In fact, for the AES, the 7- [26] and 8-round known-key distinguishers [19, Sect. 4.1] attacked correlation intractability of the round-reduced ciphers, while the 10-round distinguishers [19, Sect. 4.2] and beyond [20] are closer to breaking "indifferentiability-like" properties. The meaningfulness and influences of these two sorts of known-key models have incurred intensive discussion or even debt [19, 20].

By our results, for the natural paradigm underlying common block ciphers including the AES, the complexity for correlation intractability is 40% less than the complexity for indifferentiability. This matches the aforementioned cryptanalytic practice. While similar results have been shown with respect to the iterated Even-Mansour ciphers [8, 14], the model of CD network is more fine-grained (despite the inherently weak bounds), and we thus believe it sheds some lights on known-key attack model from the perspective of provable security.

1.4 Other Related Work

Certain models for SPNs could be proved secure against certain cryptanalytic approaches [12, 35, 33, 40]. As a variant of indifferentiability, public indifferentiability is introduced independently by Dodis et al. [15], Naito et al. [34] and Yoneyama et al. [42]. Mandal et al. [30] introduce a new and simpler variant of indifferentiability called seq-indifferentiability. Soni and Tessaro [38] introduced another form of seq-indifferentiability called *CP-sequential indifferentiability*, which restricts the distinguisher's queries to be "construction-primitive-sequential". Some other variants of indifferentiability were introduced in [2, 9] in order to formalize known-key security of block ciphers. Finally, Dodis et al. [16] shows the first positive results for the indifferentiability security of the CDNs. Based on this work, we prove that 3-round NLCDNs achieve seq-indifferentiability.

1.5 Organization

We supply necessary notations and definitions in Section 2. Then present our attack on 2-round NLCDNs in Section 3. Our main result, the seq-indifferentiability proofs for 3-round NLCDNs, is then given in Section 4. Finally, Section 5 concludes.

[2] Recall that 5 rounds are needed for NLCDNs to achieve plain indifferentiability [16].

2 Preliminaries

2.1 Notations

We write $[w]$ for the set of integers $\{1, \ldots, w\}$. We denote by bold letters, e.g., $\mathbf{x}$, bit strings of length wn, where $|\mathbf{x}|$ stands for its length. Using n-bit S-boxes, such a string $\mathbf{x}$ will be divided into w blocks, each of which is of n bits. For $i \in [w]$, the i-th n-bit block of $\mathbf{x}$ is denoted $\mathbf{x}[i]$ (i.e., $|\mathbf{x}[i]| = n$). We let $N = 2^n$ to simplify notations.

A random (invertible) permutation $\mathcal{Z} : \{+,-\} \times \{0,1\}^{wn} \to \{0,1\}^{wn}$ accepts queries of the form $(+, \mathbf{x})$ (i.e., forward queries) or $(-, \mathbf{y})$ (i.e., backward queries). As our positive result addresses a 3-round CDN with non-linear diffusion layers (NLCDN for short), we use A_j, B_j, C_j to refer to the S-boxes in the 1st, 2nd, and 3rd rounds (as sketched in Fig. 3). The idealized model of such a 3-round NLCDN relies on a tuple of $3w$ independent random permutations $\mathcal{P} = (\mathcal{P}_{A_1}, \ldots, \mathcal{P}_{A_w}, \mathcal{P}_{B_1}, \ldots, \mathcal{P}_{B_w}, \mathcal{P}_{C_1}, \ldots, \mathcal{P}_{C_w})$, where $\mathcal{P}_{\mathcal{T}_j} := \{+,-\} \times \{0,1\}^n \to \{0,1\}^n$ for every $\mathcal{T} \in \{A, B, C\}$ and every $j \in [w]$. To simplify notations, we assume that $\mathcal{P}$ provides a single interface $\mathcal{P}(\mathcal{T}_j, \delta, x)$ for all the $3w$ permutations, where $\mathcal{T}_j \in \{A_1, ..., A_w, B_1, ..., B_w, C_1, ..., C_w\}$ indicates the S-box being queried, $\delta \in \{+,-\}$ indicates the direction of the query, and $x \in \{0,1\}^n$ indicates the concrete queried value.

2.2 Confusion-Diffusion Networks

The CDN and NLCDN constructions First, we formalize *r-round confusion-diffusion networks*. Fix integers $w, n, r \in \mathbb{N}$ as parameters. Let

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

be an array of wr 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 $j \in [w]$ and will serve in the confusion layers. Given $\mathbf{x} \in \{0,1\}^{wn}$, we denote $\overline{\mathcal{P}}_i(\mathbf{x})$ as

$$\overline{\mathcal{P}}_i(\mathbf{x}) = \mathcal{P}_{i,1}(\mathbf{x}[1]) \| \mathcal{P}_{i,2}(\mathbf{x}[2]) \| ... \| \mathcal{P}_{i,w}(\mathbf{x}[w])$$

which means the i-th confusion layer. In other words, $\overline{\mathcal{P}}_i$ is a permutation of $\{0,1\}^{wn}$ and can also be defined by setting

$$\overline{\mathcal{P}}(\mathbf{x})[j] = P_{i,j}(\mathbf{x}[j]).$$

Let

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

be an arbitrary sequence or $r-1$ permutations and each of them from $\{0,1\}^{wn}$ to $\{0,1\}^{wn}$. It will be the diffusion layer, which only has certain (simple) combinatorial properties rather than sophisticated cryptographic properties.

With all the above, the function CDN is written as

$$\mathsf{CDN}^{\mathcal{P}}_{\Pi}(\mathbf{x}) = \overline{\mathcal{P}}_r(\pi_{r-1}(\ldots \overline{\mathcal{P}}_2(\pi_1(\overline{\mathcal{P}}_1(\mathbf{x}))) \ldots)) = \mathbf{y} \tag{1}$$

The value w and r will be called *width of confusion layer* and *rounds*. As mentioned, in our primary focus 3-round NLCDNs, we use A, B, C instead of $\overline{\mathcal{P}}_1, \overline{\mathcal{P}}_2, \overline{\mathcal{P}}_3$ for the S-boxes.

Combinatorial properties of the diffusion layers We now use the definitions in [16] to formalize the properties that the diffusion layers Π have to fulfill in order to result in a secure CD network. Given a vector $\mathbf{x}$ and two 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 to $\mathbf{x}[i]$ $(i \neq j)$, by replacing $\mathbf{x}[j]$ with the input $x \in \{0,1\}^n$, and by considering only the j'-th block of output. The properties are defined unidirectionally: π might satisfy a property but π^{-1} does not.

Then, the quantity $\mathsf{MaxPreEx}$ is defined as

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

Briefly, it formalizes the maximal number of $x \in \{0,1\}^n$ such that, once "extended" to a wn-bit string $\mathbf{x}$ in a pre-defined manner, the corresponding wn-bit image $\mathbf{y} = \pi(\mathbf{x})$ has at least one n-bit block equal $y \in \{0,1\}^n$. We further define

Then, the quantity $\mathsf{MaxColl}$ is defined as

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

Briefly, it formalizes the maximal number of $x, x' \in \{0,1\}^n$ such that, once "extended" to a wn-bit strings $\mathbf{x}$ and $\mathbf{x}$', the corresponding wn-bit images $\mathbf{y} = \pi(\mathbf{x})$ and $\mathbf{y}' = \pi(\mathbf{x}')$ collide on at least one n-bit block. A concrete non-linear D-box with $\mathsf{MaxPreEx}(\pi) = \mathsf{MaxColl}(\pi) \approx O(w)$ was given in [16, Appendix D].

2.3 Sequential Indifferentiability and Correlation Intractability

We first informally introduce indifferentiability, and we concentrate on CDNs to ease understanding. In this setting, a distinguisher $\mathcal{D}$ is trying to distinguish an idealized $\mathsf{CDN}^{\mathcal{P}}$ from a random wn-bit permutation $\mathcal{Z}$, with the help of the underlying random S-boxes $\mathcal{P}$. Hence, in the real world, $\mathcal{D}$ is interacting with two oracles, namely $(\mathsf{CDN}^{\mathcal{P}}, \mathcal{P})$. In the ideal world, the "position" of the non-existing oracle $\mathcal{P}$ will be filled by a simulator $\mathcal{S}^{\mathcal{Z}}$. By these, $\mathsf{CDN}^{\mathcal{P}}$ is *indifferentiable from* $\mathcal{Z}$, if there exists an efficient simulator $\mathcal{S}^{\mathcal{Z}}$ making queries to $\mathcal{Z}$, such that the ideal system $(\mathcal{Z}, \mathcal{S}^{\mathcal{Z}})$ and the real system $(\mathsf{CDN}^{\mathcal{P}}, \mathcal{P})$ are *indistinguishable* in the view of any distinguisher $\mathcal{D}$.

The sequential indifferentiability (seq-indifferentiability in short) setting also considers a distinguisher $\mathcal{D}$ trying to distinguish the ideal $(\mathcal{Z}, \mathcal{S}^{\mathcal{Z}})$ and the real $(\mathsf{CDN}^{\mathcal{P}}, \mathcal{P})$. Unlike the above (plain) indifferentiability, seq-indifferentiability focuses on *sequential distinguishers* (*seq-distinguishers* for short), i.e., a certain type of distinguishers that issue queries in a strict order. Concretely, a seq-distinguisher $\mathcal{D}$ is *primitive-construction-sequential*, if it proceeds with three

steps: (1) $\mathcal{D}$ queries the (real or ideal) construction $\mathsf{CDN}^{\mathcal{P}}$ or $\mathcal{Z}$, without querying the (real or simulated) primitive $\mathcal{S}$ or $\mathcal{S}^{\mathcal{Z}}$; (2) $\mathcal{D}$ queries the primitive $\mathcal{S}$ or $\mathcal{S}^{\mathcal{Z}}$, without querying the construction $\mathsf{CDN}^{\mathcal{P}}$ or $\mathcal{Z}$; (3) $\mathcal{D}$ outputs its decision. This order is reflected by the numbers in Fig. 2.

Using the notion of seq-distinguishers, the definition of seq-indifferentiability due to [8] is as follows.

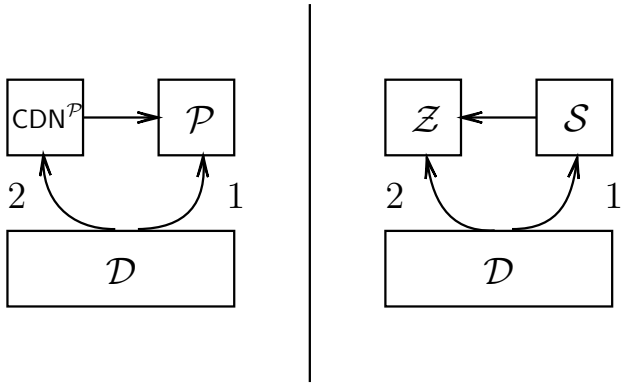

Fig. 2. The definition of sequential indifferentiability. The numbers near the arrows indicate the order of distinguisher's query. If the distinguisher first query "1", it could query "2" next. If it first query "2", it could not query "1" any more

Definition 1 (Seq-indifferentiability). *The idealized network* $\mathsf{CDN}^{\mathcal{P}}$ *with oracle access to random permutations* $\mathcal{P}$ *is statistically and strongly* $(q, \sigma, t, \varepsilon)$*-seq-indifferentiable from a random* wn*-bit permutation* $\mathcal{Z}$*, if there exists a simulator* $\mathcal{S}^{\mathcal{Z}}$ *such that for any sequential distinguisher* $\mathcal{D}$ *making at most* q *queries,* $\mathcal{S}^{\mathcal{Z}}$ *issues at most* σ *queries to* $\mathcal{Z}$ *and runs in time at most* t*, and it holds*

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

As mentioned, seq-indifferentiability already implies correlation intractability in the idealized model [30, 8]. The notion *correlation intractability* was introduced by Canetti et al. [6] to capture the feature that there is no exploitable relation between the inputs and outputs of the function ensembles in question. It was transposed to idealized models to guarantee similar feature on idealized constructions. Formally, we first give the definition (from [8]) of evasive relation.

Definition 2 (Evasive Relation). *An* m*-ary relation* $\mathcal{R}$ *over pairs of binary sequences is said* (q, ϵ)*-evasive with respect to the random* wn*-bit permutation* $\mathcal{Z}$*, if for any PPT oracle Turing machine* $\mathcal{M}$ *issuing at most* q *oracle queries, it holds*

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

Definition 3 (Correlation Intractability). *Let $\mathcal{R}$ be an m-ary relation. The idealized network $\mathsf{CDN}^{\mathcal{S}}$ with oracle access to the random S-boxes $\mathcal{S}$ is (q, ϵ)-correlation intractable with respect to $\mathcal{R}$, if for any oracle Turing machine $\mathcal{M}$ issuing at most q oracle queries, it holds*

$$\Pr\big[(x_1, \ldots, x_m) \leftarrow \mathcal{M}^{\mathcal{P}} : ((x_1, \ldots, x_m), (\mathsf{CDN}^{\mathcal{P}}(x_1), \ldots, \mathsf{CDN}^{\mathcal{P}}(x_m))) \in \mathcal{R}\big] \leq \epsilon.$$

With the above definitions, the implication of seq-indifferentiability is formally stated as follows [8].

Theorem 1. *For an idealized block cipher construction $\mathcal{C}^{\mathcal{F}}$ which has oracle access to ideal primitives $\mathcal{F}$ and makes at most c queries to $\mathcal{F}$ in total, if $\mathcal{C}^{\mathcal{F}}$ is $(q+cm, \sigma, \epsilon)$-seq-indifferentiable from another ideal primitive $\mathcal{G}$ (m is the number of binary sequences), then for any m-ary relation $\mathcal{R}$ which is $(\sigma+m, \epsilon_{\mathcal{R}})$-evasive with respect to $\mathcal{G}$, $\mathcal{C}^{\mathcal{F}}$ is $(q, \epsilon+\epsilon_{\mathcal{R}})$-correlation intractable with respect to $\mathcal{R}$.*

3 Attack 2-round CD

The attack against 2-round CDN is neither primitive-construction-sequential nor construction-primitive-sequential in [16]. In this section we exhibit a primitive-construction-sequential distinguisher against 2-round CDN making only 2 oracle queries to mitigate the gap. The assumption on the D-boxes is that it is an efficiently computable function rather than an oracle. The running time of our distinguisher may be exponential $O(2^n)$ or even $O(2^{wn})$. Though, it remains valid in the *information theoretic setting*, and confirms the *tightness* of our positive result on 3 rounds.

1. Find $b, d_1, d_2 \in \{0,1\}^n$ such that $D(b\|d_1)[1] = D(b\|d_2)[1]$;
2. Query the right oracles for $A_1^{-1}(b) \to a$, $A_2^{-1}(d_1) \to c_1$, and $A_2^{-1}(d_2) \to c_2$.
3. Query the left oracle P for $P(a\|c_1) \to f_1\|h_1$ and $P(a\|c_2) \to f_2\|h_2$, and outputs 1 if and only if $f_1 = f_2$.

If P is the 2-round CDN oracle, it necessarily holds $f_1 = f_2$ since $D(b\|d_1)[1] = D(b\|d_2)[1]$, which means the distinguisher always outputs 1. On the other hand, to simulate consistently in the ideal world, the simulator has to run ahead to find a pair of inputs/outputs $y_1 = \mathcal{Z}(+, a\|c_1)$ and $y_2 = \mathcal{Z}(+, a\|c_2)$ of the random permutation $\mathcal{Z}$ such that $y_1[1] = y_2[1]$, the probability of which is $O(q^2/2^n)$ within q queries. The distinguishing advantage is thus $1 - O(q^2/2^n) \approx 1$ for any simulator making $q \ll 2^{n/2}$ queries to P.

4 Sequential Indifferentiability of 3-round NLCDNs

The main result of this work is formally stated as follows.

Theorem 2. *Assuming that* $\mathcal{P} = (\mathcal{P}_{A_1} \dots \mathcal{P}_{A_w}, \mathcal{P}_{B_1} \dots \mathcal{P}_{B_w}, \mathcal{P}_{C_1} \dots \mathcal{P}_{C_w})$ *is a tuple of* $3w$ *independent random* n*-bit permutations, then the 3-round confusion-diffusion network with oracle access to* $\mathsf{CDN}^{\mathcal{P}}$ *is strongly and statistically* $(q, \sigma, t, \varepsilon)$*-seq-indifferentiable from a* wn*-bit random permutation* $\mathcal{Z}$*, where* $\sigma = q^w, t = O(q^w)$ *and*

$$\varepsilon = \frac{4q^w(q^w+q)}{N-q^w-q} + \frac{4w(q^w+q)^2\big(\mathsf{MaxPreEx}(\pi) + \mathsf{MaxCoPr}(\pi)\big)}{N-q^w-q} + \frac{1}{N^w}. \quad (2)$$

As mentioned in Sect. 2.2, a non-linear D-box construction with

$$\mathsf{MaxPreEx}(\pi) = \mathsf{MaxCoPr}(\pi) \approx O(w)$$

was given in [16, Appendix D]. It easy to verify that the other terms in Theorem 2 are all of the order $O(q^{2w}/N)$, which further means

$$\varepsilon = O\Big(\frac{q^{2w}}{2^n}\Big).$$

By Theorem 1, we have that for any $(q^w, \epsilon_{\mathcal{R}})$-evasive relation, the 3-round NLCDN is $(q, \epsilon_{\mathcal{R}} + O(q^{2w}/2^n))$-correlation intractable with respect to $\mathcal{R}$. We stress that $\mathsf{MaxPreEx}(\pi) = \mathsf{MaxCoPr}(\pi) \approx O(w)$ and thus the above concrete results are only achievable with *non-linear* D-boxes [16] (which is not surprising in turn).

To prove it, we: (1) build a simulator (Section 4.1); (2) bound the complexity of the simulator (Section 4.2); (3) introduce the intermediate system for the proof (Section 4.3); (4) prove that the simulator simulates well (Sections 4.4 and 4.5).

4.1 Overview of the Simulator

We follow the approach of *explicit randomness technique* of [11, 8], namely, letting the simulator $\mathcal{S}$ have explicit access to $\mathcal{P}$ and query it to obtain necessary random values. We denote by $\mathcal{S}(\mathcal{P}, \mathcal{Z})$ the simulator for 3 round CDN which access $\mathcal{P}$ (and $\mathcal{Z}$).

To keep track of previously answered queries, $\mathcal{S}$ internally maintains $3w$ tables $(A_1, \dots, A_w, B_1, \dots, B_w, C_1, \dots, C_w)$ that have entries in the form of (x, y) for $x, y \in \{0,1\}^n$. For $\mathcal{T} \in \{A, B, C\}$ and $j \in [w]$, we denote by $\mathcal{T}_j^+(x)$ the n-bit value such that $\big(x, \mathcal{T}_j^+(x)\big) \in \mathcal{T}_j$, and write $\mathcal{T}_j^+(x) = \perp$ if there is no pair of the form $(x, \star)$ in $\mathcal{T}_j$. Similarly by symmetry, we denote by $\mathcal{T}_j^-(y)$ the n-bit value such that $\big(\mathcal{T}_j^-(y), y\big) \in \mathcal{T}_j$, and write $\mathcal{T}_j^-(y) = \perp$ once no such pair exists. For $\delta \in \{+, -\}$, we denote by $\bar{\delta}$ the opposite of δ. For example, when $\delta = +$, $\mathcal{T}_j^{\bar{\delta}}$ refers to $\mathcal{T}_j^-$.

The basic idea is Coron et al.'s simulation via *chain completion technique* [11], which has achieved succes in (weaker) indifferentiability proofs of a variety of idealized block ciphers. It requires the simulator $\mathcal{S}$ to *detect* "partial" computation chains formed by the queries of the distinguisher, and *completes* the chains

in advance by querying the random permutation $\mathcal{Z}$, so that $\mathcal{S}$ is ready for answering queries in the future. To simulate answers that are consistent with $\mathcal{Z}$, $\mathcal{S}$ has to use the answer from $\mathcal{Z}$ to define some simulated answers: this action is called *adaptation*. Specifically, our simulator views every tuple of w queries to the (2nd round) S-boxes $B_1, ..., B_w$ as a partial chain, and completes it by defining entries in $A_1, ..., A_w$ or $C_1, ..., C_w$ depending on the context, as depicted in Fig. 3.

$\mathcal{S}$ offers an interface Query$(\mathcal{T}_j, \delta, x)$ to the distinguisher (which is the same as the interface of $\mathcal{P}$), where $\mathcal{T} \in \{A, B, C\}$ and $j \in [w]$ indicate the concrete S-box being queried, $\delta \in \{+, -\}$ indicates whether this a direct of inverse query, and $x \in \{0,1\}^n$ is the actual queried value. Upon a query Query$(\mathcal{T}_j, \delta, x)$, $\mathcal{S}$ checks the table $\mathcal{T}_j$ to see whether the corresponding answer $\mathcal{T}_j^{\delta}(x)$ is already defined. When this is the case, it returns $\mathcal{T}_j^{\delta}(x)$ to finish this response. Otherwise, it draws a random response $y \leftarrow \mathcal{P}(\mathcal{T}_j, \delta, x)$ from the random permutation $\mathcal{P}$ and invokes a private procedure SetTable$(\mathcal{T}_j^{\delta}, x, y)$. The latter procedure adds (x, y) to $\mathcal{T}_j$.

Then, if $= \mathcal{B}$, $\mathcal{S}$ invokes another private procedure AdaptC (resp. AdaptA) if $\delta = +$ (resp. $\delta = -$) to complete detected partial chains as mentioned before. In detail, when $\delta = +$, then for every $\mathbf{x}^B[j] = x$, $\mathcal{S}$ calls AdaptC, which further computes $\mathbf{x}^C = \pi_2(\mathbf{y}^B)$, $\mathbf{x}^A = \mathit{Block}(A, -, \pi_1^{-1}(\mathbf{x}^B))$,[4] and queries $\mathcal{Z}(-, \mathbf{x}^A) \rightarrow \mathbf{y}^C$. The procedure AdaptC then adapts: for $j = 1, ..., w$, it defines $(\mathbf{x}^C[j], \mathbf{y}^C[j])$ as a new entry of the table C_j. Entries to-be-adapted may cause inconsistency when an entry of the form $(\mathbf{x}^C[j], \star)$ or $(\star, \mathbf{y}^C[j])$ already exists in C_j. In this case, our simulator *overwrites* the existing entries and breaks the bijectivity of the partially defined maps. This is the major source of inconsistency, and its unlikeness constitutes a main intermediate goal of our remaining proofs. The procedure AdaptA is similar to the above by symmetry. The chain completion strategy is illustrated in Fig. 3. $\mathcal{S}$ eventually returns $\mathcal{T}_j^{\delta}(x)$ as the response. This means queries of the form (A_j, δ, x) or (C_j, δ, x) won't trigger chain detection, and are simply answered with randomness from $\mathcal{P}$. The formal description in pseudocode is given in Algorithm 1.

4.2 Simulator Efficiency

As the first step, we must prove that the complexity of the simulator $\mathcal{S}$ is polynomial in q.

Lemma 1. *If the simulator receives at most q queries in total, then for every $j \in [w]$, the tables $A_1, \ldots, A_w, B_1, \ldots, B_w, C_1, \ldots, C_w$ of $\mathcal{S}$ has $|B_j| \leq q$, $\left|A_j\right| \leq q^w + q$, and $\left|C_j\right| \leq q^w + q$. The simulator executes AdaptA and AdaptC for at most q^w times, makes at most q^w queries to $\mathcal{Z}$ and runs in time $O(q^w)$.*

Proof. For $j \in [w]$, it is clear that $\left|B_j\right|$ only increases by at most 1 when the distinguisher makes a query to Query(B_j, δ, x), and thus $\left|B_j\right| \leq q$. On the other hand, $\left|A_j\right|$ may increase in two cases:

[4] The private procedure Block$(\mathcal{T}, \delta, \mathbf{t})$ computes a complete S-box layer on the input $\mathbf{t} \in \{0,1\}^{wn}$, where $\mathcal{T} \in \{A, B, C\}$ and $\delta \in \{+, -\}$.

Algorithm 1 Simulator $\mathcal{S}(\mathcal{Z}, \mathcal{P})$

procedure Query($\mathcal{T}_j, \delta, x$)
  if $\mathcal{T}_j^\delta(x) = \perp$ **then**
    $y \leftarrow \mathcal{P}(\mathcal{T}_j, \delta, x)$
    SetTable($\mathcal{T}_j, x, y$)
    if $\mathcal{T} = B$ **and** $\delta = +$ **then**
      forall $\mathbf{x}^B, \mathbf{y}^B$ **s.t.** $\mathbf{x}^B[j] = x$ **do**
        AdaptC($\mathbf{x}^B, \mathbf{y}^B$)
    if $\mathcal{T} = B$ **and** $\delta = -$ **then**
      forall $\mathbf{x}^B, \mathbf{y}^B$ **s.t.** $\mathbf{y}^B[j] = x$ **do**
        AdpatA($\mathbf{x}^B, \mathbf{y}^B$)
  return $\mathcal{T}_j^\delta(x)$

private procedure AdaptC($\mathbf{x}^B, \mathbf{y}^B$)
  $\mathbf{x}^C = \pi_2(\mathbf{y}^B)$
  $\mathbf{x}^A = \text{Block}(A, -, \pi_1^{-1}(\mathbf{x}^B))$
  $\mathbf{y}^C = \mathcal{Z}(+, \mathbf{x}^A)$
  forall $j \in \{1, \ldots, w\}$ **do**
    SetTable($C_j, \mathbf{x}^C[j], \mathbf{y}^C[j]$)

private procedure AdaptA($\mathbf{x}^B, \mathbf{y}^B$)
  $\mathbf{y}^A = \pi_1^{-1}(\mathbf{y}^B)$
  $\mathbf{y}^C = \text{Block}(C, +, \pi_2(\mathbf{y}^B))$
  $\mathbf{x}^A = \mathcal{Z}(-, \mathbf{y}^C)$
  forall $j \in \{1, \ldots, w\}$ **do**
    SetTable($A_j, \mathbf{x}^A[j], \mathbf{y}^A[j]$)

procedure SetTable($\mathcal{T}_j^\delta, x, y$)
  $\mathcal{T}_j^\delta(x) \leftarrow y$
  $\mathcal{T}_j^\delta(y) \leftarrow x$

private procedure Block($\mathcal{T}, \delta, \mathbf{t}$)
  forall $j \in \{1, \ldots, w\}$ **do**
    if $\mathcal{T}_j^\delta(\mathbf{t}[j]) = \perp$ **then**
      $\mathbf{u}[j] \leftarrow \mathcal{P}(\mathcal{T}_j, \delta, \mathbf{t}[j])$
      SetTable($\mathcal{T}_j, \mathbf{t}[j], \mathbf{u}[j]$)
    $\mathbf{u}[j] \leftarrow \mathcal{T}_j^\delta(\mathbf{t}[j])$
  return u

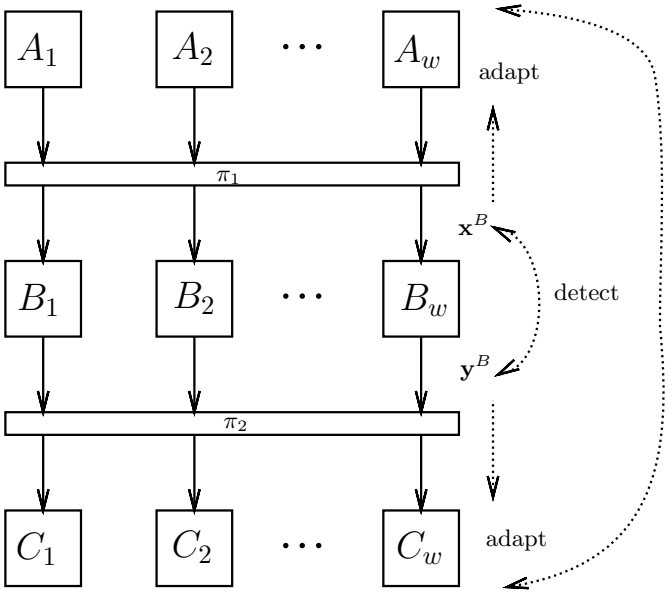

Fig. 3. Our simulation strategy.

(1) The distinguisher makes a query to Query(A_j, δ, x), and
(2) $\mathcal{S}$ executes AdaptA($\mathbf{x}^B, \mathbf{y}^B$).

The procedure AdaptA is executed once for every wn-bit "combined" string $\mathbf{x}^B \in B_1 \times \cdots \times B_w$ detected by $\mathcal{S}$. Therefore, the number of executions is at most q^w. This plus the increment due to the q adversarial queries yield $\left|A_j\right| \leq q^w + q$. The argument for $\left|C_j\right| \leq q^w + q$ is similar by symmetry. Then, each execution of AdaptA/AdaptC makes 1 query to $\mathcal{Z}$, which establishes the q^w query complexity. Finally, the simulator computations are clearly dominated by the executions of AdaptA/AdaptC, and this establishes the $O(q^w)$ time complexity. □

4.3 Intermediate Systems

We follow [8] and use three games to facilitate the proof (see Fig. 4). The game G_1 captures the interaction between the distinguisher and the ideal world $(\mathcal{Z}, \mathcal{S}(\mathcal{Z}, \mathcal{P}))$. $\mathcal{Z}$ is a wn-bit random permutation and $\mathcal{P}$ is a tuple of n-bit independent random permutation $(\mathcal{P}_{A_1} \ldots \mathcal{P}_{A_w}, \mathcal{P}_{B_1} \ldots \mathcal{P}_{B_w}, \mathcal{P}_{C_1} \ldots \mathcal{P}_{C_w})$, plays the role of S-boxes in CDN which is mentioned in Section 2.3. The simulator $\mathcal{S}(\mathcal{Z}, \mathcal{P})$ has access to both $\mathcal{Z}$ and $\mathcal{P}$. Our rules for constructing game strictly follow the rules constructed in [1, 8], and all use random permutation $\mathcal{P}$ as source of randomness. The game G_3 captures interaction between the distinguisher and the real world $(\mathsf{CDN}^{\mathcal{P}}, \mathcal{P})$. We construct a intermediate system G_2. It lies between G_1 and G_3 and functions as a bridge to simplify the proof. The intermediate game G_2 captures the interaction between the distinguisher and the system $(\mathsf{CDN}^{\mathcal{S}(\mathcal{Z},\mathcal{P})}, \mathcal{S}(\mathcal{Z}, \mathcal{P}))$, i.e., it is modified from G_1 by replacing $\mathcal{Z}$ with the CDN construction. In other words, the right oracle is the simulator $\mathcal{S}(\mathcal{Z}, \mathcal{P})$ with oracle access to random permutation $\mathcal{Z}$, but now the left oracle is CDN construction with oracle access to $\mathcal{S}(\mathcal{Z}, \mathcal{P})$.

4.4 Probability of Overwriting

As mentioned before, during executing the procedures AdaptA and AdaptC, our simulator may overwrite already defined entries and cause inconsistency. In this section we show this event of overwriting, in fact, happens with a bounded probability.

The event overwriting only occurs during the execution of SetTable. We begin by considering the probability of line 4 and line 34. These lines only cause overwriting when the sampled values collide with the value previously added by AdaptA or AdaptC. Since the size of A_j and C_j is $q^w + q$ by Lemma 1, the obtained $y \leftarrow \mathcal{P}(\mathcal{T}_j, \delta, x)$ is uniform in at least $N - q^w - q$ possibilities. The probability that y already exists is $\frac{2(q^w+q)}{N-q^w-q}$ since there are at most $q^w + q$ random assignment in tables A_j and C_j. The procedure AdaptA, resp. AdaptC, is executed by at most q^w times. By these, we have

$$\Pr\left[\textit{line } 4,\ 34\ \textit{overwrite}\right] \leq \frac{2q^w(q^w+q)}{N-q^w-q}. \tag{3}$$

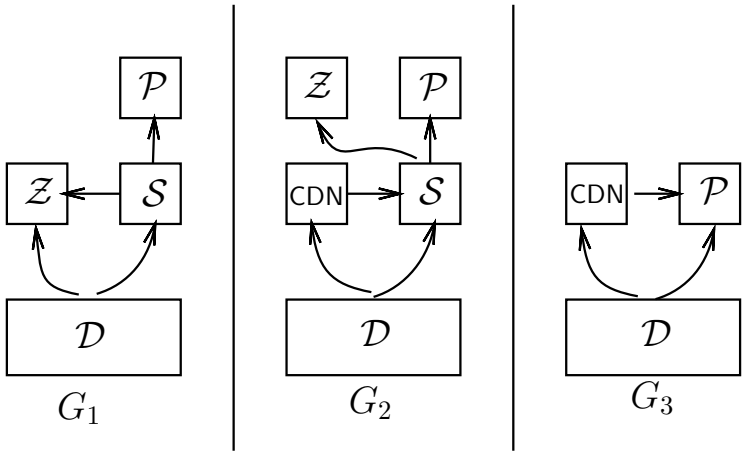

Fig. 4. Games and the involved primitives used in our proof.

Then, we consider the overwriting in AdaptC or AdaptA. During executing the former AdaptC, the occurrence of overwriting is due to $C_j(x) \neq \perp$ or $C_j^{-1}(y) \neq \perp$. Assume that table C_j already have k pairs ($k \leq q^w + q$) $(\mathbf{x}_1^C, \mathbf{y}_1^C), \ldots, (\mathbf{x}_k^C, \mathbf{y}_k^C)$ before this execution. By construction (line 14), we define $\mathbf{x}_i^C = \pi_2(\mathbf{y}_i^B)$ has chance to cause the following two types of overwriting:

- PreEx: $\mathbf{x}_i^C[j] \in \mathcal{C}_j$, $(1 \leq i \leq k, 1 \leq j \leq w)$
- Coll: $\mathbf{x}_i^C[j] = \mathbf{x}_{i'}^C[j]$, $(1 \leq i \leq i' \leq k, 1 \leq j \leq w)$

where $\mathcal{C}_j$ is represented as the domain of table C_j. In other words, $\mathcal{C}_j = \{x \in \{0,1\}^n : C_j(x) \neq \perp\}$. It is clear that PreEx and Coll includes all the possibilities of bad events in AdaptC. By Lemma 1, the size of $\mathcal{C}_j$ is at most $q^2 + q$. We first discuss the probability of occurrence of PreEx. As mentioned before, we denoted $\pi_{j,j'}^{\mathbf{x}}$ be the function from $\{0,1\}^n$ to $\{0,1\}^n$. It represents in the D-boxes, we split the input x and output y into the j-th and j'-th block. We now define:

$$\mathsf{MaxPreEx}(\pi_2) = \max_{\mathbf{x},j,h,y} \left|\{x \in \{0,1\}^n : \pi_{j,h}^{\mathbf{x}}(x) = y\}\right|.$$

Since the size of C_j is at most $q^w + q$, $\mathbf{x}_i^C$ is uniformly random in a set of size at least $N - q^w - q$, So we can find the probability of occurrence of PreEx for $\mathbf{x}_i^C[j] = \pi_2(\mathbf{y}^B)$ at most:

$$\frac{\mathsf{MaxPreEx}(\pi_2)|\mathcal{C}_j|}{N - q^w - q}.$$

For all $\mathbf{x}^C$, the probability would be at most:

$$\Pr\left[\prod_{i=1}^{k}\prod_{j=1}^{w}\mathsf{PreEx}\right] \leq \frac{wk(q^w+q)\mathsf{MaxPreEx}(\pi_2)}{N - q^w - q}. \tag{4}$$

Next, we consider the probability of Coll. Coll occurs if and only if $\mathbf{x}_i^C[j] = \mathbf{x}_{i'}^C[j]$. There are two different situations here: if $\mathbf{x}_i^C$, $\mathbf{x}_{i'}^C$ are from distinct calls, the probability of $\mathbf{x}_i^C[j] = \mathbf{x}_{i'}^C[j]$ is at most:

$$\frac{\mathsf{MaxPreEx}(\pi_2)}{N - q^w - q}.$$

In this case, the value range of bad event has only one value. $|\mathcal{C}_j|$ is replaced by 1. If $\mathbf{x}_i^C$, $\mathbf{x}_{i'}^C$ are from the same calls, the probability of $\mathbf{x}_i^C[j] = \mathbf{x}_{i'}^C[j]$ is at most:

$$\frac{\mathsf{MaxColl}(\pi_2)}{N - q^w - q},$$

which we define that:

$$\mathsf{MaxColl}(\pi_2) = \max_{\mathbf{x} \neq \mathbf{x}', j, h} \left| \{x \in \{0,1\}^n : \pi_{j,h}^{\mathbf{x}}(x) = \pi_{j,h}^{\mathbf{x}'}(x)\} \right|.$$

We let $\mathsf{MaxCoPr}(\pi) = \max(\mathsf{MaxPreEx}(\pi), \mathsf{MaxColl}(\pi))$, thus:

$$\Pr\left[\prod_{i=1}^{k}\prod_{i'=1}^{k}\prod_{j=1}^{w}\mathsf{Coll}\right] \leq \frac{wk^2\mathsf{MaxCoPr}(\pi_2)}{N - q^w - q}. \tag{5}$$

Gathering Eqs. (4) and (5), and using $k \leq q^w + q$, the probability to have overwriting due to executing AdaptC is bounded by

$$\begin{aligned}\Pr[\text{AdaptC } \textit{overwrites}] &\leq \frac{wk(q^w+q)\mathsf{MaxPreEx}(\pi_2)}{N - q^w - q} + \frac{wk^2\mathsf{MaxCoPr}(\pi_2)}{N - q^w - q} \\ &= \frac{w(q^w+q)^2\big(\mathsf{MaxPreEx}(\pi_2) + \mathsf{MaxCoPr}(\pi_2)\big)}{N - q^w - q}.\end{aligned} \tag{6}$$

Similar reasoning holds for AdaptA executions by symmetry, giving rise to the same bound

$$\begin{aligned}\Pr[\text{AdaptA } \textit{overwrites}] &\leq \frac{wk(q^w+q)\mathsf{MaxPreEx}(\pi_1)}{N - q^w - q} + \frac{wk^2\mathsf{MaxCoPr}(\pi_1)}{N - q^w - q} \\ &= \frac{w(q^w+q)^2\big(\mathsf{MaxPreEx}(\pi_1) + \mathsf{MaxCoPr}(\pi_1)\big)}{N - q^w - q}.\end{aligned} \tag{7}$$

Gathering Eqs. (3) and (7), we evetually have the probability of overwriting.

$$\Pr\big[\mathit{Overwriting}\big] \leq \frac{2q^w(q^w+q)}{N - q^w - q} + \frac{2w(q^w+q)^2\big(\mathsf{MaxPreEx}(\pi) + \mathsf{MaxCoPr}(\pi)\big)}{N - q^w - q}, \tag{8}$$

where $\mathsf{MaxPreEx}(\pi)$, $\mathsf{MaxColl}(\pi)$, and $\mathsf{MaxCoPr}(\pi)$ stand for the maximal quantity among the two diffusion layers π_1, π_2, i.e.,

$$\begin{aligned}\mathsf{MaxPreEx}(\pi) &= \max\big(\mathsf{MaxPreEx}(\pi_1), \ldots, \mathsf{MaxPreEx}(\pi_2^{-1})\big) \\ \mathsf{MaxColl}(\pi) &= \max\big(\mathsf{MaxColl}(\pi_1), \ldots, \mathsf{MaxColl}(\pi_2^{-1})\big) \\ \mathsf{MaxCoPr}(\pi) &= \max\big(\mathsf{MaxCoPr}(\pi_1), \ldots, \mathsf{MaxCoPr}(\pi_2^{-1})\big).\end{aligned}$$

4.5 Statistical Distance Between Games

In this section, we will complete the final step of the proof. Recall from Section 4.3 that we built three games to imitate real world and ideal world. First, we consider the transition from G_1 to G_2. Note that both G_1 and G_2 has the same pair $(\mathcal{Z}, \mathcal{P})$, $\mathcal{Z}$ is the random wn-bit permutation and $\mathcal{P}$ is a tuple of random permutations $\mathcal{P}_{\mathcal{T}_j}$. The pair is *bad*, if the simulator overwrites an entry of the table $\mathcal{T}_j$, specifically, A_j, C_j during G_2; otherwise, the pair is *good*.

We first address the statistical distance between G_1 and G_2.

Lemma 2. *For any distinguisher $\mathcal{D}$ making at most q queries, the statistical distance between G_1 and G_2 is bounded by*

$$\left| \Pr\left[\mathcal{D}^{G_1(\mathcal{S}(\mathcal{Z},\mathcal{P}),\mathcal{Z})} = 1\right] - \Pr\left[\mathcal{D}^{G_2(\mathcal{S}(\mathcal{Z},\mathcal{P}),\mathsf{CDN}^{\mathcal{S}(\mathcal{Z},\mathcal{P})})}\right] \right| \leq \Pr\left[(\mathcal{Z},\mathcal{P}) \textit{ is bad}\right].$$

Proof. Since the distinguisher is sequential in the sense of Definition 1, in G_1 and G_2, it necessarily first queries $\mathcal{S}(\mathcal{Z},\mathcal{P})$ and then $\mathcal{Z}$ (in G_1) or $\mathsf{CDN}^{\mathcal{S}(\mathcal{Z},\mathcal{P})}$ (in G_2) only. If the pair is good, the answers $\mathcal{D}$ received from G_1 and G_2 are the same since they stem from the same randomness source. On the other side, $\mathcal{Z}$ is an ideal primitive and CDN is the structure that exists in the real state, they could not trigger bad event. So, the statistical distance between G_1 and G_2 is determined by pair $(\mathcal{Z},\mathcal{P})$ and will not be greater than the pair $(\mathcal{Z},\mathcal{P})$ is bad. Bad event will not triggered by $\mathcal{S}$ unless the pair $(\mathcal{Z}, \mathcal{P})$ is bad. Hence, the statistical distance between G_1 and G_2 is actually the probability of bad events. □

Next, we consider the transition from G_2 and G_3, i.e. the transition from $(\mathcal{Z},\mathcal{P})$ to $\mathcal{P}$ which is the most important part. Thus, we use the *randomness mapping argument* of Holenstein et al. [24]. In detail, we define a map Γ on tuples of random permutations $(\mathcal{Z},\mathcal{P})$. When the pair $(\mathcal{Z},\mathcal{P})$ is bad, $\Gamma(\mathcal{Z},\mathcal{P}) = \perp$ which is a special symbol. Otherwise, $\Gamma(\mathcal{Z},\mathcal{P})$ is the tuple of $3w$ tables $\beta = (\beta_1, ..., \beta_{3w})$ standing at the end of the execution $G_2(\mathcal{Z},\mathcal{P})$. It is easy to see such tables $\beta = (\beta_1, ..., \beta_{3w})$ defines $3w$ partial permutations and a partial permutation is a function $\beta_i : \{+,-\} \times \{0,1\}^n \to \{0,1\}^n \cup \{*\}$ such that for all $x, y \in \{0,1\}^n, \beta_i(+,x) = y \neq * \Leftrightarrow \beta_i(-,y) = x \neq *$. The map Γ is defined for good pairs $(\mathcal{Z},\mathcal{P})$ as follows: run $\mathcal{D}^{G_2(\mathcal{Z},\mathcal{P})}$, and consider the tables $\mathcal{T}_j$ of the $\mathcal{S}$ at the end of the execution; then fill all undefined entries of the $\mathcal{T}_j$ with the special symbol *.

We say that a tuple of permutation $\mathcal{P}$ extends a tuple of partial permutation $\beta = (\beta_1, ..., \beta_{3w})$, denoted $\mathcal{P} \vdash \beta$, if for each β_i and $\mathcal{P}$ agree on all entries such that $\beta_i(\delta, x) \neq *$. By the definition of the randomness mapping, for any good tuple of partial permutation β, the output of $\mathcal{D}^{G_2(\mathcal{Z},\mathcal{P})}$ and $\mathcal{D}^{G_3(\mathcal{P})}$ are equal for any pair $(\mathcal{Z},\mathcal{P})$ such that $\Gamma(\mathcal{Z},\mathcal{P}) = \beta$ and any tuple of permutations $\mathcal{P}$ such that $\mathcal{P} \vdash \beta$. We can conclude that for all β, the distance $\Delta(G_2, G_3)$ between G_2

and G_3 is bounded by

$$\begin{aligned}\Delta(G_2, G_3) = &\left| \Pr\left[\mathcal{D}^{G_2(\mathcal{Z},\mathcal{P})} = 1\right] - |\Pr\left[\mathcal{D}^{G_3(\mathcal{P})} = 1\right]\right| \\ \leq &\ \Pr[(\mathcal{Z}, \mathcal{P}) \textit{ is bad}] + \sum \Pr[\Gamma(\mathcal{Z}, \mathcal{P}) = \beta] - \sum \Pr[\mathcal{P} \vdash \beta]. \quad (9)\end{aligned}$$

For $\mathcal{D}^{G_3(\mathcal{P})}$, let $\bar{q}_{\mathcal{T}_j}$ be the good execution of $\mathcal{P} \vdash \beta$, then:

$$\Pr[\mathcal{P} \vdash \beta] = \prod_{\mathcal{T}} \prod_{j=0}^{w} \prod_{l=0}^{|\bar{q}_{\mathcal{T}_j}|-1} \frac{1}{N-l}. \quad (10)$$

For $\mathcal{D}^{G_2(\mathcal{Z},\mathcal{P})}$, let $\bar{p}_{\mathcal{T}_j}$ be the good pair of $\Gamma(\mathcal{Z}, \mathcal{P}) = \beta$, then:

$$\Pr[\Gamma(\mathcal{Z}, \mathcal{P}) = \beta] = \left(\prod_{l=0}^{|\mathcal{Z}|-1} \frac{1}{N^w - l} \right) \cdot \left(\prod_{\mathcal{T}} \prod_{j=1}^{w} \prod_{l=0}^{|\bar{p}_{\mathcal{T}_j}|-1} \frac{1}{N-l} \right). \quad (11)$$

Lemma 3. *Under the conditions of (10) and (11), for $\mathcal{T} \in \{B\}$, $|\bar{p}_{\mathcal{T}_j}| = |\bar{q}_{\mathcal{T}_j}|$, and for $\mathcal{T} \in \{A, C\}$, if there exist two non-negative integers a,c such that $a+c = |\mathcal{Z}|$, then $|\bar{q}_{A_j}| = |\bar{p}_{A_j}| + a$, $|\bar{q}_{C_j}| = |\bar{p}_{C_j}| + c$*

Proof. Recall that G_3 is the real world, $|\bar{q}_{\mathcal{T}_j}| = |\mathcal{T}_j|$ since there is no adapt mechanism in it. In G_2, $|B_j|$ will never be adapted, so $|\bar{p}_{\mathcal{T}_j}| = |\mathcal{T}_j| = |\bar{q}_{\mathcal{T}_j}|$ if $\mathcal{T} \in \{B\}$. $|A_j|$ and $|C_j|$ is adapted when the simulator call procedures AdaptA or AdaptC. Noted that $\mathcal{Z}$ is only called by AdaptA or AdaptC, so the times AdaptA or AdaptC called is equal to the size of table $\mathcal{Z}$. Assume that AdaptA is called a times and AdaptC is called c times, so clearly $|\mathcal{Z}| = a + c$. Due to the adapt mechanism of G_2, $|\bar{q}_{A_j}| = |\bar{p}_{A_j}| + a$, $|\bar{q}_{C_j}| = |\bar{p}_{C_j}| + c$. □

We divide (10) by (11), and apply Lemma 3:

$$\begin{aligned}\frac{\Pr[\mathcal{P} \vdash \beta]}{\Pr[\Gamma(\mathcal{Z}, \mathcal{P}) = \beta]} &= \frac{\prod_{\mathcal{T}} \prod_{j=0}^{w} \prod_{h=0}^{|\bar{q}_{\mathcal{T}_j}|-1} \frac{1}{N-h}}{\left(\prod_{h=0}^{|\mathcal{Z}|-1} \frac{1}{N^w-h} \right) \cdot \left(\prod_{\mathcal{T}} \prod_{j=1}^{w} \prod_{h=0}^{|\bar{p}_{\mathcal{T}_j}|-1} \frac{1}{N-h} \right)} \\ &\geq \prod_{h=0}^{a-1} \frac{1}{(N-h)^w} \cdot \prod_{h=0}^{c-1} \frac{1}{(N-h)^w} \cdot \prod_{h=0}^{a+c-1} (N^w - h) \\ &= \frac{N^w - 1}{N^w} \cdot \prod_{h=1}^{a-1} \frac{1}{(N-h)^w} \cdot \prod_{h=1}^{c-1} \frac{1}{(N-h)^w} \cdot \prod_{h=2}^{a+c-1} (N^w - h) \\ &= \frac{N^w - 1}{N^w} \cdot \frac{\prod_{h=a}^{a+c-1} (N^w - h)}{\prod_{h=1}^{c-1} (N-h)^w} \\ &\geq \frac{N^w - 1}{N^w} = 1 - \frac{1}{N^w}.\end{aligned} \quad (12)$$

Gathering Eqs. (12) and (9), we have

$$\begin{aligned}\Delta(G_2, G_3) &\leq \Pr[(\mathcal{Z}, \mathcal{P}) \; is \; bad] + \sum \Pr[\Gamma(\mathcal{Z}, \mathcal{P}) = \beta] - \sum \Pr[\mathcal{P} \vdash \beta] \\ &= \Pr[(\mathcal{Z}, \mathcal{P}) \; is \; bad] + \sum \Pr[\Gamma(\mathcal{Z}, \mathcal{P}) = \beta] \left(1 - \frac{\Pr[\Gamma(\mathcal{Z}, \mathcal{P}) = \beta]}{\Pr[\mathcal{P} \vdash \beta]}\right) \\ &\leq \Pr[(\mathcal{Z}, \mathcal{P}) \; is \; bad] + \sum \Pr[\Gamma(\mathcal{Z}, \mathcal{P}) = \beta] \cdot \frac{1}{N^w} \\ &\leq \Pr[(\mathcal{Z}, \mathcal{P}) \; is \; bad] + \frac{1}{N^w}. \end{aligned} \tag{13}$$

Using Lemma 3 again, we eventually have Eq. (2).

$$\begin{aligned}&\left| \Pr\left[\mathcal{D}^{G_1(\mathcal{Z}, \mathcal{P})} = 1\right] - |\Pr\left[\mathcal{D}^{G_3(\mathcal{P})} = 1\right] \right| \\ &\leq \Delta(G_2, G_3) + \Pr[(\mathcal{Z}, \mathcal{P}) \; is \; bad] \\ &= 2 \Pr[(\mathcal{Z}, \mathcal{P}) \; is \; bad] + \frac{1}{N^w} \\ &= \frac{4q^w(q^w + q)}{N - q^w - q} + \frac{4w(q^w + q)^2 \big(\mathsf{MaxPreEx}(\pi) + \mathsf{MaxCoPr}(\pi)\big)}{N - q^w - q} + \frac{1}{N^w}. \end{aligned} \tag{14}$$

5 Conclusion

We characterize the sequential indifferentiability of Confusion-Diffusion Networks (CDNs). Assuming using random permutations as S-boxes and non-linear permutations as the diffusion layer, we exhibit a sequential distinguisher against 2-round CDNs (strengthening Dodis et al.'s negative result [16]) and prove sequential indifferentiability for 3-round CDN. Non-linear D-boxes satisfy certain combinatorial requirements, and this is crucial for the proof of Section 4.4. This was also central for the full indifferentiability results of [16]: as mentioned in our Introduction, using non-linear D-boxes 5 rounds are proved indifferentiable, while 9 rounds are needed for linear D-boxes. Hence, to achieve sequential indifferentiability, the exact number of rounds required by non-linear CDNs is 3, which is better than that (5 rounds) needed for full indifferentiability. These complement Dodis et al.'s results in the full indifferentiability setting [16] and deepen the theory of known-key security of block ciphers.

Acknowledgments

We sincerely appreciate the anonymous reviewers for their insightful feedback that helps us improving our presentations greatly. Chun Guo was partly supported by the Program of Taishan Young Scholars of the Shandong Province, the Program of Qilu Young Scholars (Grant No. 61580089963177) of Shandong University, the National Natural Science Foundation of China (Grant No. 62002202), and the Shandong Nature Science Foundation of China (Grant No. ZR2020MF053).

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.) Advances in Cryptology—Crypto. LNCS, vol. 8042, pp. 531–550. Springer (Aug 2013). https://doi.org/10.1007/978-3-642-40041-4_29
2. Andreeva, E., Bogdanov, A., Mennink, B.: Towards understanding the known-key security of block ciphers. In: Moriai, S. (ed.) Fast Software Encryption (FSE). LNCS, vol. 8424, pp. 348–366. Springer (Mar 2014). https://doi.org/10.1007/978-3-662-43933-3_18
3. Barbosa, M., Farshim, P.: Indifferentiable authenticated encryption. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part I. LNCS, vol. 10991, pp. 187–220. Springer (Aug 2018). https://doi.org/10.1007/978-3-319-96884-1_7
4. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: On the indifferentiability of the sponge construction. In: Smart, N.P. (ed.) Advances in Cryptology—Eurocrypt. LNCS, vol. 4965, pp. 181–197. Springer (Apr 2008). https://doi.org/10.1007/978-3-540-78967-3_11
5. Bertoni, G., Peeters, M., Van Assche, G., et al.: The keccak reference (2011)
6. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited (preliminary version). In: 30th Annual ACM Symposium on Theory of Computing (STOC). pp. 209–218. ACM Press (May 1998). https://doi.org/10.1145/276698.276741
7. Cogliati, B., Dodis, Y., Katz, J., Lee, J., Steinberger, J.P., Thiruvengadam, A., Zhang, Z.: Provable security of (tweakable) block ciphers based on substitution-permutation networks. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part I. LNCS, vol. 10991, pp. 722–753. Springer (Aug 2018). https://doi.org/10.1007/978-3-319-96884-1_24
8. Cogliati, B., Seurin, Y.: On the provable security of the iterated Even-Mansour cipher against related-key and chosen-key attacks. In: Oswald, E., Fischlin, M. (eds.) Advances in Cryptology—Eurocrypt 2015, Part I. LNCS, vol. 9056, pp. 584–613. Springer (Apr 2015). https://doi.org/10.1007/978-3-662-46800-5_23
9. Cogliati, B., Seurin, Y.: Strengthening the known-key security notion for block ciphers. In: Peyrin, T. (ed.) Fast Software Encryption (FSE). LNCS, vol. 9783, pp. 494–513. Springer (Mar 2016). https://doi.org/10.1007/978-3-662-52993-5_25
10. Coron, J.S., Dodis, Y., Malinaud, C., Puniya, P.: Merkle-Damgård revisited: How to construct a hash function. In: Shoup, V. (ed.) Advances in Cryptology—Crypto. LNCS, vol. 3621, pp. 430–448. Springer (Aug 2005). https://doi.org/10.1007/11535218_26
11. Coron, J.S., Holenstein, T., Künzler, R., Patarin, J., Seurin, Y., Tessaro, S.: How to build an ideal cipher: The indifferentiability of the Feistel construction. Journal of Cryptology **29**(1), 61–114 (Jan 2016). https://doi.org/10.1007/s00145-014-9189-6
12. Daemen, J., Rijmen, V.: The wide trail design strategy. In: Honary, B. (ed.) 8th IMA International Conference on Cryptography and Coding. LNCS, vol. 2260, pp. 222–238. Springer, Heidelberg (Dec 2001)
13. Daemen, J., Rijmen, V.: The design of Rijndael, vol. 2. Springer (2002)
14. Dai, Y., Seurin, Y., Steinberger, J.P., Thiruvengadam, A.: Indifferentiability of iterated Even-Mansour ciphers with non-idealized key-schedules: Five rounds are necessary and sufficient. In: Katz, J., Shacham, H. (eds.) Advances in Cryptology—Crypto 2017, Part III. LNCS, vol. 10403, pp. 524–555. Springer (Aug 2017). https://doi.org/10.1007/978-3-319-63697-9_18

15. Dodis, Y., Ristenpart, T., Shrimpton, T.: Salvaging Merkle-Damgård for practical applications. In: Joux, A. (ed.) Advances in Cryptology—Eurocrypt. LNCS, vol. 5479, pp. 371–388. Springer (Apr 2009). https://doi.org/10.1007/978-3-642-01001-9_22
16. Dodis, Y., Stam, M., Steinberger, J.P., Liu, T.: Indifferentiability of confusion-diffusion networks. In: Fischlin, M., Coron, J.S. (eds.) Advances in Cryptology—Eurocrypt. LNCS, vol. 9666, pp. 679–704. Springer (May 2016). https://doi.org/10.1007/978-3-662-49896-5_24
17. Feistel, H., Notz, W.A., Smith, J.L.: Cryptographic techniques for machine to machine data communications. IBM Thomas J. Watson Research Center (1971)
18. Gao, Y., Guo, C., Wang, M., Wang, W., Wen, J.: Beyond-birthday-bound security for 4-round linear substitution-permutation networks. IACR Transactions on Symmetric Cryptology pp. 305–326 (2020). https://doi.org//10.13154/tosc.v2020.i3.305-326
19. Gilbert, H.: A simplified representation of AES. In: Sarkar, P., Iwata, T. (eds.) Advances in Cryptology—Asiacrypt. LNCS, vol. 8873, pp. 200–222. Springer (Dec 2014). https://doi.org/10.1007/978-3-662-45611-8_11
20. Grassi, L., Rechberger, C.: Revisiting Gilbert's known-key distinguisher. Des. Codes Cryptogr. **88**(7), 1401–1445 (2020). https://doi.org/10.1007/s10623-020-00756-5, https://doi.org/10.1007/s10623-020-00756-5
21. Guo, C., Katz, J., Wang, X., Yu, Y.: Efficient and secure multiparty computation from fixed-key block ciphers. In: 2020 IEEE Symposium on Security and Privacy. pp. 825–841. IEEE Computer Society Press, San Francisco, CA, USA (May 18–21, 2020). https://doi.org/10.1109/SP40000.2020.00016
22. Guo, C., Lin, D.: A synthetic indifferentiability analysis of interleaved double-key Even-Mansour ciphers. In: Iwata, T., Cheon, J.H. (eds.) Advances in Cryptology—Asiacrypt. LNCS, vol. 9453, pp. 389–410. Springer (Nov / Dec 2015). https://doi.org/10.1007/978-3-662-48800-3_16
23. Guo, C., Lin, D.: Separating invertible key derivations from non-invertible ones: sequential indifferentiability of 3-round even–mansour. Designs, Codes and Cryptography **81**(1), 109–129 (2016). https://doi.org/10.1007/s10623-015-0132-0
24. 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.) 43rd ACM STOC. pp. 89–98. ACM Press (Jun 2011). https://doi.org/10.1145/1993636.1993650
25. Iwata, T., Kurosawa, K.: On the pseudorandomness of the AES finalists - RC6 and Serpent. In: Schneier, B. (ed.) Fast Software Encryption (FSE). LNCS, vol. 1978, pp. 231–243. Springer (Apr 2001). https://doi.org/10.1007/3-540-44706-7_16
26. Knudsen, L.R., Rijmen, V.: Known-key distinguishers for some block ciphers. In: Kurosawa, K. (ed.) Advances in Cryptology—Asiacrypt. LNCS, vol. 4833, pp. 315–324. Springer (Dec 2007). https://doi.org/10.1007/978-3-540-76900-2_19
27. Lai, X.: On the design and security of block ciphers. Ph.D. thesis, ETH Zurich (1992)
28. Lai, X., Massey, J.L.: A proposal for a new block encryption standard. In: Damgård, I. (ed.) Advances in Cryptology—Eurocrypt. LNCS, vol. 473, pp. 389–404. Springer (May 1991). https://doi.org/10.1007/3-540-46877-3_35
29. Liu, Y., Rijmen, V., Leander, G.: Nonlinear diffusion layers. Designs, Codes and Cryptography **86**(11), 2469–2484 (2018). https://doi.org/10.1007/s10623-018-0458-5

30. Mandal, A., Patarin, J., Seurin, Y.: On the public indifferentiability and correlation intractability of the 6-round Feistel construction. In: Cramer, R. (ed.) Theory of Cryptography Conference 2012. LNCS, vol. 7194, pp. 285–302. Springer (Mar 2012). https://doi.org/10.1007/978-3-642-28914-9_16
31. Maurer, U., Renner, R.: From indifferentiability to constructive cryptography (and back). In: Hirt, M., Smith, A.D. (eds.) TCC 2016-B, Part I. LNCS, vol. 9985, pp. 3–24. Springer (Oct / Nov 2016). https://doi.org/10.1007/978-3-662-53641-4_1
32. Maurer, U.M., Renner, R., Holenstein, C.: Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In: Naor, M. (ed.) Theory of Cryptography Conference. LNCS, vol. 2951, pp. 21–39. Springer (Feb 2004). https://doi.org/10.1007/978-3-540-24638-1_2
33. Miles, E., Viola, E.: Substitution-permutation networks, pseudorandom functions, and natural proofs. In: Safavi-Naini, R., Canetti, R. (eds.) Advances in Cryptology—Crypto. LNCS, vol. 7417, pp. 68–85. Springer (Aug 2012). https://doi.org/10.1007/978-3-642-32009-5_5
34. Naito, Y., Yoneyama, K., Wang, L., Ohta, K.: How to confirm cryptosystems security: The original Merkle-Damgård is still alive! In: Matsui, M. (ed.) Advances in Cryptology—Asiacrypt. LNCS, vol. 5912, pp. 382–398. Springer (Dec 2009). https://doi.org/10.1007/978-3-642-10366-7_23
35. Park, S., Sung, S.H., Lee, S., Lim, J.: Improving the upper bound on the maximum differential and the maximum linear Hull probability for SPN structures and AES. In: Johansson, T. (ed.) Fast Software Encryption (FSE). LNCS, vol. 2887, pp. 247–260. Springer (Feb 2003). https://doi.org/10.1007/978-3-540-39887-5_19
36. Rogaway, P., Steinberger, J.P.: Constructing cryptographic hash functions from fixed-key blockciphers. In: Wagner, D. (ed.) Advances in Cryptology—Crypto. LNCS, vol. 5157, pp. 433–450. Springer (Aug 2008). https://doi.org/10.1007/978-3-540-85174-5_24
37. Shannon, C.E.: Communication theory of secrecy systems. Bell Systems Technical Journal **28**(4), 656–715 (1949)
38. Soni, P., Tessaro, S.: Public-seed pseudorandom permutations. In: Coron, J., Nielsen, J.B. (eds.) Advances in Cryptology—Eurocrypt 2017, Part II. LNCS, vol. 10211, pp. 412–441. Springer (Apr / May 2017). https://doi.org/10.1007/978-3-319-56614-6_14
39. Soni, P., Tessaro, S.: Naor-Reingold goes public: The complexity of known-key security. In: Nielsen, J.B., Rijmen, V. (eds.) Advances in Cryptology—Eurocrypt 2018, Part III. LNCS, vol. 10822, pp. 653–684. Springer (Apr / May 2018). https://doi.org/10.1007/978-3-319-78372-7_21
40. Sun, B., Liu, M., Guo, J., Rijmen, V., Li, R.: Provable security evaluation of structures against impossible differential and zero correlation linear cryptanalysis. In: Fischlin, M., Coron, J.S. (eds.) Advances in Cryptology—Eurocrypt. LNCS, vol. 9665, pp. 196–213. Springer (May 2016). https://doi.org/10.1007/978-3-662-49890-3_8
41. Wu, W., Zhang, L.: LBlock: A lightweight block cipher. In: Lopez, J., Tsudik, G. (eds.) Intl. Conference on Applied Cryptography and Network Security (ACNS). LNCS, vol. 6715, pp. 327–344. Springer (Jun 2011). https://doi.org/10.1007/978-3-642-21554-4_19
42. Yoneyama, K., Miyagawa, S., Ohta, K.: Leaky random oracle (extended abstract). In: Baek, J., Bao, F., Chen, K., Lai, X. (eds.) ProvSec 2008. LNCS, vol. 5324, pp. 226–240. Springer, Heidelberg (Oct / Nov 2008)

43. Zhang, W., Bao, Z., Lin, D., Rijmen, V., Yang, B., Verbauwhede, I.: Rectangle: a bit-slice lightweight block cipher suitable for multiple platforms. Science China Information Sciences **58**(12), 1–15 (2015)