IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2020, No. 2, pp. 68–92. DOI:10.13154/tosc.v2020.i2.68-92

# Beyond-Birthday-Bound Secure Cryptographic Permutations from Ideal Ciphers with Long Keys

Ryota Nakamichi and Tetsu Iwata

Nagoya University, Nagoya, Japan
r_nakami@echo.nuee.nagoya-u.ac.jp,tetsu.iwata@nagoya-u.jp

**Abstract.** Coron et al. showed a construction of a 3-round $2n$-bit cryptographic permutation from three independent $n$-bit ideal ciphers with $n$-bit keys (TCC 2010). Guo and Lin showed a construction of a $(2d-1)$-round $dn$-bit cryptographic permutation from $2d-1$ independent $n$-bit ideal ciphers with $\kappa n$-bit keys, where $d = \kappa + 1$ (Cryptography and Communications, 2015). These constructions have an indifferentiability security bound of $O(q^2/2^n)$ against adversaries that make at most $q$ queries. The bound is commonly referred to as birthday-bound security.
In this paper, we show that a 5-round version of Coron et al.'s construction and $(2d+1)$-round version of Guo and Lin's construction yield a cryptographic permutation with an indifferentiability security bound of $O(q^2/2^{2n})$, i.e., by adding two more rounds, these constructions have beyond-birthday-bound security. Furthermore, under the assumption that $q \leq 2^n$, we show that Guo and Lin's construction with $2d+2\ell-1$ rounds yields a cryptographic permutation with a security bound of $O(q^2/2^{(\ell+1)n})$, where $1 \leq \ell \leq d-1$, i.e., the security bound exponentially improves by adding every two more rounds, up to $4d-3$ rounds. To the best of our knowledge, our result gives the *first* cryptographic permutation that is built from $n$-bit ideal ciphers and has a full $n$-bit indifferentiability security bound.

**Keywords:** Cryptographic permutation · Ideal cipher · Provable security · Indifferentiability · Coefficient-H technique

## 1 Introduction

**Background.** A cryptographic permutation is a non-keyed public permutation that is designed to behave like a public random permutation, and it is the core primitive of permutation-based cryptography, e.g., it can be used as a primitive to construct hash functions, encryption schemes, message authentication codes, and authenticated encryption schemes [BDPA08, BDPA10, BDPA11, ADMA15, BDH+17]. Designing a secure and efficient cryptographic permutation is an important problem in symmetric cryptography, and a large number of dedicated designs have been proposed, e.g., we have the permutation in SHA-3 [NIS14] and 384-bit permutation Gimli [BKL+17].

For this problem, Coron et al. [CDMS10] initiated constructing a cryptographic permutation by using block ciphers. In [CDMS10], a 3-round construction of a $2n$-bit cryptographic permutation from three independent $n$-bit block ciphers with $n$-bit keys (described in Fig. 1(a)) is proposed, and its security was shown in the indifferentiability framework introduced by Maurer et al. [MRH04]. Specifically, it was proved that the 3-round construction using publicly accessible random block ciphers (modeled as independent ideal ciphers) is indifferentiable from a $2n$-bit random permutation, and the 2-round version cannot achieve indifferentiability. The security bound of adversaries that make at most $q_c$ construction queries and at most $q_p$ primitive queries is $O(q^2/2^n)$ for $q = q_c + q_p$, and this security bound is often referred to as birthday-bound security (with respect to $n$, the


Received: 2020-03-02, Accepted: 2020-05-01, Published: 2020-07-24

block length of the ideal cipher). This implies that a $2n$-bit random permutation can be securely replaced by the 3-round construction provided that $q \ll 2^{n/2}$ holds. In [CDMS10], it was also proved that if we apply three $n$-bit ideal ciphers with longer $(n+k)$-bit keys to the similar iterative construction, then it gives a $2n$-bit keyed permutation with $k$-bit keys (described in Fig. 1(b)), and it is indifferentiable from a $2n$-bit ideal cipher with $k$-bit keys with a security bound of $O(q^2/2^n)$. This construction is a domain extender for the ideal cipher that doubles the domain, and the variant with $k = 0$ is the construction of the cryptographic permutation in Fig. 1(a).

Guo and Lin [GL15] improved the domain extender for the ideal cipher of [CDMS10], and proposed an iterative construction that extends the domain by a factor of $d \geq 2$. A variant of the construction in [GL15] by setting the key length as $k = 0$ provides a cryptographic permutation. Specifically, for a positive integer $\kappa$, [GL15] provides an iterative construction of a $dn$-bit cryptographic permutation by using $n$-bit block ciphers with $\kappa n$-bit keys (modeled as independent ideal ciphers), where $d = \kappa + 1$ and the number of rounds $r$ is equal to the number of block ciphers used, which is $r = 2d - 1$ (described in Fig. 1(c) for $r$ rounds). From the result in [GL15], the $(2d-1)$-round construction is indifferentiable from a $dn$-bit random permutation with a security bound of $O(q^2/2^n)$, and [GL15] also shows that $(2d-2)$-round version cannot achieve indifferentiability.

We see that the indifferentiability security bounds of these constructions are limited to birthday-bound security, and a very natural question is whether we can have a stronger security bound by increasing the number of rounds. In this paper, we call the security bound that guarantees beyond $q = q_c + q_p \approx 2^{n/2}$ queries beyond-birthday-bound security (BBB security). A construction of a cryptographic permutation from $n$-bit ideal ciphers with BBB security remains as an open question.

**Our Results.** We study the problem of constructing a secure cryptographic permutation from block ciphers modeled as ideal ciphers in the provable security paradigm. We present the *first* BBB security proof (with respect to $n$, the block length of the ideal cipher) as a construction of cryptographic permutations. We prove under the assumption $q = q_c + q_p \leq 2^n$ that the iterative construction in Fig. 1(c) is indifferentiable from a $dn$-bit random permutation with an indifferentiability security bound of $O(q^2/2^{(\ell+1)n})$ for $r = 2d + 2\ell - 1$, where $1 \leq \ell \leq d - 1$ is an integer.

This implies that the 5-round version of Coron et al.'s construction and $(2d+1)$-round version of Guo and Lin's construction have an indifferentiability security bound of $O(q^2/2^{2n})$, i.e., by adding two more rounds, these constructions have BBB security. Furthermore, under the assumption that $q \leq 2^n$, our result shows that the security bound of Guo and Lin's construction exponentially improves by adding every two more rounds, up to $4d - 3$ rounds.

From the technical side, our BBB security proof is made possible by designing a simulator that is tailored to handle various collisions between $n$-bit random variables. That is, when we define a bad event in our security proof for the $(2d + 2\ell - 1)$-round construction, all the events are defined so that they involve collisions between $(\ell + 1)n$-bit random variables, which is the main difference from the birthday-bound security proofs in [CDMS10, GL15].

We emphasize that in this paper, we use BBB security to mean that the construction remains secure beyond $q = q_c + q_p \approx 2^{n/2}$ queries with respect to $n$, the block length of the underlying primitive, which is the output length and is *not* the input length. Table 1 summaries the results in [CDMS10], [GL15], and this paper.

**Implication.** We present implication of our results with practical parameters.

- If we model AES-128 [DR02] as the 128-bit ideal cipher with 128-bit keys, then the result in [CDMS10] shows that the 3-round version gives a 256-bit cryptographic

**Table 1:** Summary of our result and results in [CDMS10] and [GL15]. Here, $\kappa$, $d$, and $\ell$ are positive integers with $d = \kappa + 1$ and $1 \le \ell \le d - 1$, and $q = q_c + q_p$. $(k, n)$ in the column of block ciphers denotes $n$-bit block ciphers with $k$-bit keys and the bounds neglect constants.

| Length | Block ciphers | Rounds | Bound | Secure $q$ | Paper |
|---|---|---|---|---|---|
| $2n$ | $(n, n)$ | $r \le 2$ | 1 | insecure | [CDMS10] |
| $2n$ | $(n, n)$ | 3 | $q^2/2^n$ | $q \ll 2^{n/2}$ | [CDMS10] |
| $2n$ | $(n, n)$ | 5 | $q^2/2^{2n}$ | $q \ll 2^n$ | This paper |
| $dn$ | $(\kappa n, n)$ | $r \le 2d - 2$ | 1 | insecure | [GL15] |
| $dn$ | $(\kappa n, n)$ | $2d - 1$ | $q^2/2^n$ | $q \ll 2^{n/2}$ | [GL15] |
| $dn$ | $(\kappa n, n)$ | $2d + 1$ | $q^2/2^{2n}$ | $q \ll 2^n$ | This paper |
| $dn$ | $(\kappa n, n)$ | $2d + 2\ell - 1$ | $q^2/2^{(\ell+1)n}$ | $q \le 2^n$ | This paper |

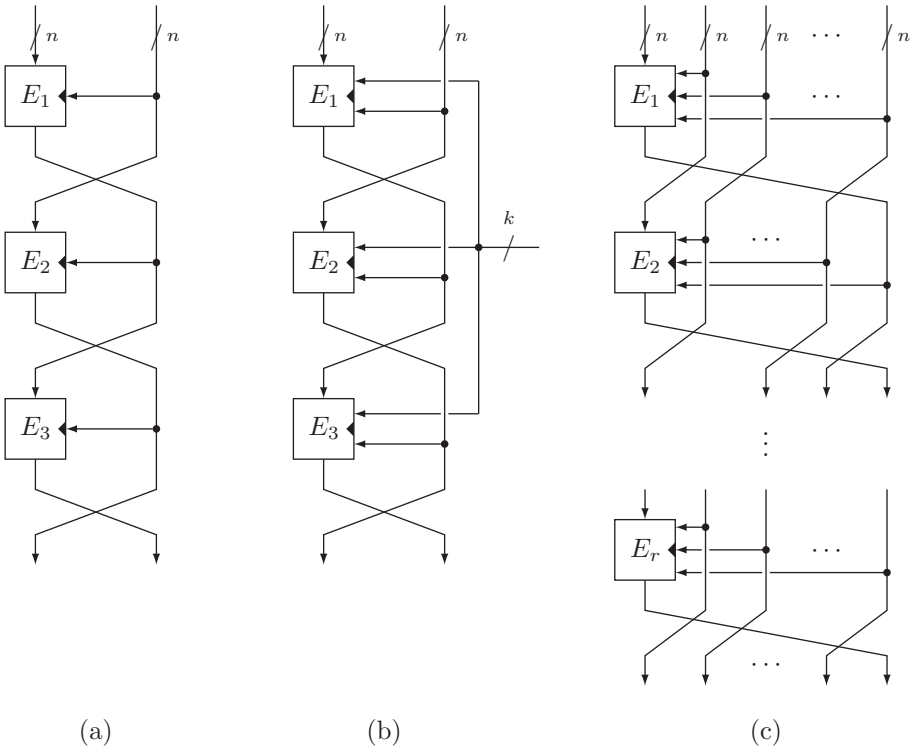



**Figure 1:** The constructions of previous works and the construction we study in this paper. (a) and (b) are Coron et al.'s constructions [CDMS10], and (c) for $r = 2d - 1$ is Guo and Lin's construction (and $k = 0$) [GL15]. We prove the security of (c) for $r = 2d + 2\ell - 1$ and $1 \le \ell \le d - 1$.

permutation with a security bound of $O(q^2/2^{128})$, whereas our result shows that the 5-round version has a security bound of $O(q^2/2^{256})$.[1]

- If we model AES-256 [DR02] as the 128-bit ideal cipher with 256-bit keys, then [GL15]

[1] We note that AES-128 has to be regarded as three or five independent ideal ciphers, and thus it has to be somehow "tweaked." A similar comment applies to AES-256 and SKINNY-128-384.

shows that the 5-round version gives a 384-bit cryptographic permutation with a security bound of $O(q^2/2^{128})$, whereas our result shows that the 7-round version has a security bound of $O(q^2/2^{256})$. Our result also shows that if $q \leq 2^{128}$, then the security bound of the 9-round construction is $O(q^2/2^{384})$.

- If we model SKINNY-128-384 [BJK+16] as the 128-bit ideal cipher with 384-bit keys, then [GL15] shows that the 7-round version gives a 512-bit cryptographic permutation with a security bound of $O(q^2/2^{128})$. Our result shows that the security bound of the 9-round construction is $O(q^2/2^{256})$, the 11-round version is $O(q^2/2^{384})$, and the 13-round version is $O(q^2/2^{512})$, where the last two bounds assume $q \leq 2^{128}$.

We emphasize that we chose the above examples to illustrate practical parameters of block and key lengths, and we are *not* proposing the instantiation. As a matter of fact, from the efficiency view point, our result does not give competitive constructions. Furthermore, our result (and those in [CDMS10, GL15]) relies on the fact that the ideal ciphers are independent. That is, even if we model AES-128, AES-256, and SKINNY-128-384 as ideal ciphers, this does not directly give us cryptographic permutations. These block ciphers have to be somehow "tweaked" so that they can be modeled as independent ideal ciphers. This implies that cryptanalyses are needed on the tweaked block ciphers, and does not allow the direct use of well-scrutinized primitives. We also remark that AES-256 was shown not to behave like an ideal cipher [BKN09, BK09], and we again remark that we use the above examples only to illustrate practical parameters.

**Related Work.** There has been a long line of research to investigating the indifferentiability of Feistel structures that use independent random oracles as round functions, see e.g., [CPS08, HKT11, MPS12, CHK+16, DS16, DKT16]. The indifferentiability framework has been used to analyze the security of various other constructions, see e.g., a line of research analyzing indifferentiability security of key-alternating ciphers [CS15, DSST17].

Minematsu [Min15] and Nakamichi and Iwata [NI19] analyzed the security of closely related constructions in the indistinguishability framework, i.e., the underlying primitive has a secret key and the adversary does not have oracle access to it. Our result can be seen as the indifferentiability counterpart of their results. Compared to [NI19], the analysis in [NI19] is more involved in that they study smaller number of rounds, while the primitive queries are absent.

The constructions in [Min15, NI19] can be seen as a construction of a block cipher that has a secret key, and there are constructions to handle various input lengths in the indistinguishability framework, see e.g., [ST13, CLMP17, BLN18, CMN18, DN18].

As mentioned above, our result does not give practically competitive cryptographic permutations in efficiency. However, see [BLLN19] for an attempt to apply the result of Coron et al. [CDMS10] to obtain efficient authenticated encryption schemes.

## 2 Preliminaries

**Notation.** For a positive integer $n$, $\{0,1\}^n$ denotes the set of all bit strings of length $n$ bits. For two bit strings $x$ and $y$, $x \,\|\, y$ is the concatenation of $x$ and $y$. For two positive integers $a$ and $b$ with $a \leq b$, we let $[a..b] = \{a, a+1, \ldots, b\}$, and for $b-a+1$ strings $X^a, X^{a+1}, \ldots, X^b \in \{0,1\}^n$ of length $n$ bits, we let $X^{[a..b]} = X^a \,\|\, X^{a+1} \,\|\, \cdots \,\|\, X^b$. For a finite set $\mathcal{S}$, $s \xleftarrow{\$} \mathcal{S}$ is the process of uniformly random selection of an element from $\mathcal{S}$ and assigning it to a variable $s$.

**Cryptographic Permutations and Block Ciphers.** Let $\mathrm{Perm}(n)$ denote the set of all permutations on $\{0,1\}^n$. A cryptographic permutation is a non-keyed public permutation

$\Phi : \{0,1\}^n \to \{0,1\}^n$, where $n$ is the (fixed) length and $\Phi(\cdot) \in \mathrm{Perm}(n)$. The inverse permutation is denoted by $\Phi^{-1}(\cdot)$, where for any $X \in \{0,1\}^n$, it holds that $X = \Phi^{-1}(\Phi(X))$. We say that $\Phi$ is an $n$-bit cryptographic permutation if $\Phi(\cdot) \in \mathrm{Perm}(n)$.

A block cipher is a keyed permutation $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$, where $k$ is the key length and $n$ is the block length. For any $K \in \{0,1\}^k$, we have $E(K,\cdot) \in \mathrm{Perm}(n)$. The ciphertext $C \in \{0,1\}^n$ for a key $K \in \{0,1\}^k$ and a plaintext $M \in \{0,1\}^n$ is $C = E(K,M)$. The decryption function is denoted by $E^{-1}(\cdot,\cdot)$, where for any $K \in \{0,1\}^k$ and $M \in \{0,1\}^n$, $M = E^{-1}(K, E(K,M))$. We say that $E$ is an $n$-bit block cipher with $k$-bit keys if $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$.

A random permutation $\pi : \{0,1\}^n \to \{0,1\}^n$ models a random cryptographic permutation, and it is defined as $\pi \xleftarrow{\$} \mathrm{Perm}(n)$. An ideal cipher $P : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ models a random block cipher, and it is defined as $P(K,\cdot) \xleftarrow{\$} \mathrm{Perm}(n)$ for any $K \in \{0,1\}^k$. That is, for each key $K \in \{0,1\}^k$, $P(K,\cdot)$ is an independent random permutation. We say that $P$ is an $n$-bit ideal cipher with $k$-bit keys if $P : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$.

**Security Definitions and Coefficient-H Technique.** In this paper, we prove the indifferentiability [MRH04] of a cryptographic permutation that uses ideal ciphers from a random permutation. It is defined as follows:

**Definition 1** ([MRH04])**.** A cryptographic permutation that uses ideal ciphers is said to be $(q_c, q_p, \epsilon)$-indifferentiable from a random permutation, if there exists a simulator $S$ such that for any adversary $\mathcal{A}$ that makes at most $q_c$ queries to a construction oracle $\mathcal{O}_c$ and $q_p$ queries to a primitive oracle $\mathcal{O}_p$, it holds that

$$|\Pr[\mathcal{A}^{\Phi^P,P} = 1] - \Pr[\mathcal{A}^{\Pi,S^\Pi} = 1]| \le \epsilon\,,$$

where $\Phi$, $P$, $\Pi$, and $S$ are the oracles of the cryptographic permutation construction and its inverse permutation, the ideal ciphers and their decryption functions, the random permutation and its inverse permutation, and the simulator, respectively. Here, $(\mathcal{O}_c, \mathcal{O}_p) \in \{(\Phi, P), (\Pi, S)\}$, $\Phi$ can make queries to $P$, and $S$ can make queries to $\Pi$.

A query by $\mathcal{A}$ to a construction oracle $\mathcal{O}_c$ is called a construction query, and a query to a primitive oracle $\mathcal{O}_p$ is called a primitive query.

Our security proof relies on the Coefficient-H technique by Patarin [Pat08] and its refinement by Chen and Steinberger [CS14]. We follow [CS14] and $P$ leaks some of the internal variables of the cryptographic permutation to $\mathcal{A}$, and we define $S$ to eliminate the obvious discrepancy.

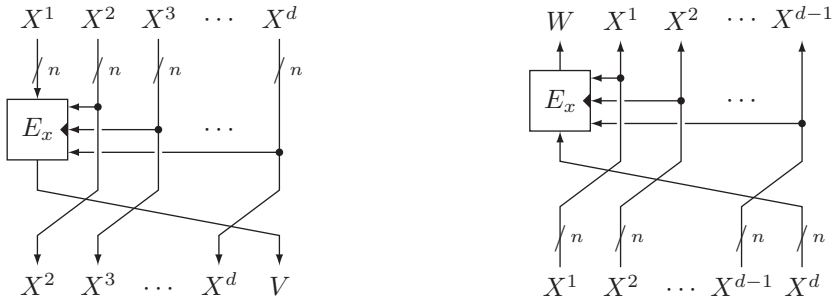
Since $\mathcal{A}$ makes at most $q_c$ queries to $\mathcal{O}_c$ and at most $q_p$ queries to $\mathcal{O}_p$, we can define a transcript $\tau$ that summarizes all query-response tuples seen by $\mathcal{A}$ during its interaction with $(\mathcal{O}_c, \mathcal{O}_p) \in \{(\Phi, P), (\Pi, S)\}$. We denote by $T^{\mathrm{re}}$ (resp. $T^{\mathrm{id}}$) the probability distribution of transcripts when $\mathcal{A}$ interacts with $(\Phi, P)$ (resp. $(\Pi, S)$). We call a transcript $\tau$ *attainable* if $\Pr[T^{\mathrm{id}} = \tau] > 0$ holds, i.e., if $\tau$ can be obtained with interacting $(\Pi, S)$. Then, the Coefficient-H technique is the following lemma.

**Lemma 1.** *Consider a deterministic adversary $\mathcal{A}$ and the set of all attainable transcripts $\mathcal{T}^{\mathrm{all}}$. Let $\mathcal{T}^{\mathrm{bad}}$ be the subset of $\mathcal{T}^{\mathrm{all}}$ with all "bad" transcripts, and $\mathcal{T}^{\mathrm{good}} = \mathcal{T}^{\mathrm{all}} \setminus \mathcal{T}^{\mathrm{bad}}$. Suppose that there exists $0 \le \epsilon_1 \le 1$ such that*

$$\frac{\Pr[T^{\mathrm{re}} = \tau]}{\Pr[T^{\mathrm{id}} = \tau]} \ge 1 - \epsilon_1$$

*holds for all $\tau \in \mathcal{T}^{\mathrm{good}}$, and there exists $0 \le \epsilon_2 \le 1$ such that $\Pr[T^{\mathrm{id}} \in \mathcal{T}^{\mathrm{bad}}] \le \epsilon_2$. Then we have*

$$|\Pr[\mathcal{A}^{\Phi^P,P} = 1] - \Pr[\mathcal{A}^{\Pi,S^\Pi} = 1]| \le \epsilon_1 + \epsilon_2\,.$$




(a) (b)

**Figure 2:** (a) $\varphi[E_x](X^{[1..d]}) = X^{[2..d]} \,\|\, V$ and (b) $\varphi^{-1}[E_x](X^{[1..d]}) = W \,\|\, X^{[1..d-1]}$

## 3 $dn$-bit Cryptographic Permutation $\Phi_r$

Fix $r \geq 1$ and $\kappa \geq 1$. We define the $r$-round algorithm $\Phi_r$ of a $dn$-bit cryptographic permutation for $d = \kappa + 1$ that uses $r$ independent $n$-bit block ciphers $E_1, \ldots, E_r$ with $\kappa n$-bit keys as

$$\Phi_r[E_1, \ldots, E_r](\bar{X}) = \varphi[E_r] \circ \varphi[E_{r-1}] \circ \cdots \circ \varphi[E_1](\bar{X}) \,,$$

where $\bar{X} \in \{0,1\}^{dn}$ is an input, and each round $\varphi[E_x] : \{0,1\}^{dn} \to \{0,1\}^{dn}$ for $x \in [1..r]$ is defined as

$$\varphi[E_x](X^{[1..d]}) = X^{[2..d]} \,\|\, V \,,$$

where $X^{[1..d]} \in \{0,1\}^{dn}$ is an input and $V = E_x(X^{[2..d]}, X^1)$. See Fig. 2(a). That is, for an input $\bar{X}$, $\Phi_r[E_1, \ldots, E_r]$ successively applies $\varphi[E_1], \ldots, \varphi[E_r]$ on $\bar{X}$. We will omit $E_1, \ldots, E_r$ and write $\Phi_r$ instead of $\Phi_r[E_1, \ldots, E_r]$ if $E_1, \ldots, E_r$ are clear from the context. $\Phi_r$ is described in Fig. 3.

The inverse algorithm of $\Phi_r[E_1, \ldots, E_r]$, denoted by $\Phi_r^{-1}[E_1, \ldots, E_r]$, is defined as

$$\Phi_r^{-1}[E_1, \ldots, E_r](\bar{X}) = \varphi^{-1}[E_1] \circ \varphi^{-1}[E_2] \circ \cdots \circ \varphi^{-1}[E_r](\bar{X}) \,,$$

where $\bar{X} \in \{0,1\}^{dn}$ is an input and $\varphi^{-1}[E_x]$ for $x \in [1..r]$ is the inverse permutation of $\varphi[E_x]$ that is naturally defined by using the decryption of $E_x$, denoted by $E_x^{-1}$, as

$$\varphi^{-1}[E_x](X^{[1..d]}) = W \,\|\, X^{[1..d-1]} \,,$$

where $X^{[1..d]} \in \{0,1\}^{dn}$ is an input and $W = E_x^{-1}(X^{[1..d-1]}, X^d)$. See Fig. 2(b). Note that for $X^{[1..d]} \in \{0,1\}^{dn}$, $X^{[2..d]}$ is used as a key in $E_x$ while $X^{[1..d-1]}$ is used as a key in $E_x^{-1}$. This notation is convenient in our security proof.

## 4 The Indifferentiability of $\Phi_{2d+2\ell-1}$ for $q \leq 2^n$

We present the following main theorem of this paper.

**Theorem 1.** *Fix $d \geq 2$ and $\ell \in [1..d-1]$. For $x \in [1..2d+2\ell-1]$, let $P_x : \{0,1\}^{(d-1)n} \times \{0,1\}^n \to \{0,1\}^n$ be an $n$-bit ideal cipher with $(d-1)n$-bit keys, and consider $\Phi_{2d+2\ell-1} = \Phi_{2d+2\ell-1}[P_1, \ldots, P_{2d+2\ell-1}]$. Then for any $\mathcal{A}$ that makes at most $q_c$ construction queries and at most $q_p$ primitive queries, where $q_c + q_p \leq 2^n$, $\Phi_{2d+2\ell-1}$ is $(q_c, q_p, \epsilon)$-indifferentiable from a $dn$-bit random permutation, where*

$$\epsilon = \frac{(q_c + q_p)^2}{2^{(\ell+1)n}} + \frac{\ell(q_c + q_p)^2}{2^{dn}} = O\left(\frac{(q_c + q_p)^2}{2^{(\ell+1)n}}\right) . \tag{1}$$

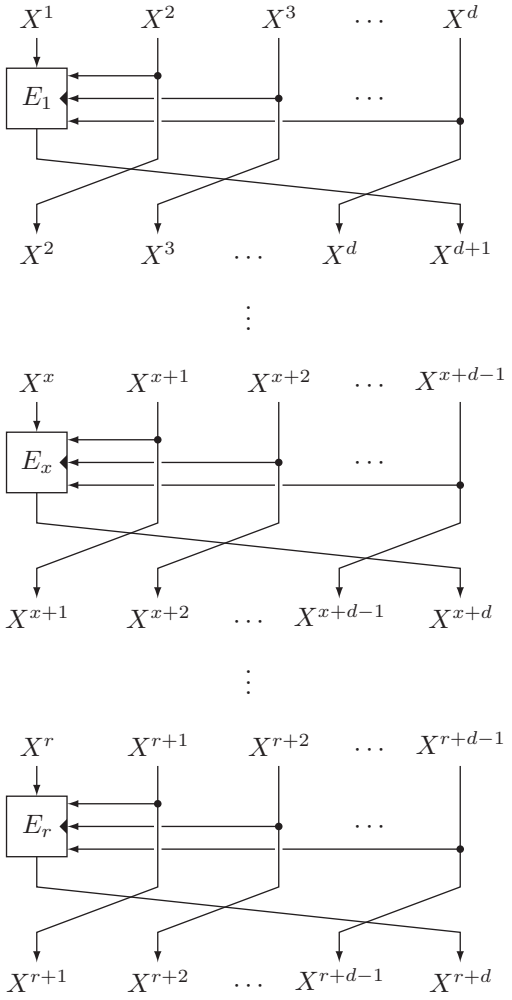



**Figure 3:** $\Phi_r[E_1, \ldots, E_r]$

**Proof Overview.** The full proof of Theorem 1 is presented in Sect. 5, and we present a proof overview here. The overall proof strategy is that we give *all the internal variables* of $\Phi_{2d+2\ell-1}$ to the adversary through *primitive* queries. To do so, whenever the adversary makes a construction query, we force the adversary to make a primitive query immediately after the construction query. When the oracle receives the primitive query, we modify it so that the oracle invokes all the internal ideal ciphers $P_1, \ldots, P_{2d+2\ell-1}$ to compute the internal variables, and returns all the internal variables to the adversary.

If our adversary makes $q_c$ construction queries and $q_p$ primitive queries, the new adversary makes $q_c$ construction queries and $q_c + q_p$ primitive queries, and we prove Theorem 1 against adversaries that make these extra primitive queries.

We then define the real world oracles, the oracles for $\Phi_{2d+2\ell-1}$ (and its inverse permutation) and $P_1, \ldots, P_{2d+2\ell-1}$ (and their decryption functions) to execute the behavior outlined above in a natural way.

The ideal world oracles, the oracle for a random permutation $\pi$ (and its inverse $\pi^{-1}$) is defined in a natural way. The simulator $S$ simulates $P_1, \ldots, P_{2d+2\ell-1}$ (and their decryption functions) with the lazy-sampling approach (see, e.g., [BR04]). To define the simulator, we introduce a concept of *upper queries* and *lower queries* for primitive queries. It is

often the case that the description of the oracle makes a distinction between encryption queries and decryption queries. We do follow this and we describe the procedure of the simulator by making a distinction between encryption/decryption queries, however, the essential distinction is made depending on upper/lower queries in that for upper queries, we can uniformly describe the procedure of the simulator, and the same holds for lower queries. In more detail, observe that $P_1, \ldots, P_{2d+2\ell-1}$ lists all the internal ideal ciphers, and $X^1, \ldots, X^{3d+2\ell-1}$ lists all the internal variables. Our simulation works as follow:

- For upper queries, we simulate $P_1, \ldots, P_{d+2\ell-1}$ through lazy-sampling to define $X^1, \ldots, X^{2d+2\ell-1}$, and we use the $dn$-bit random permutation $\pi$ to define the remaining internal variables, i.e., $X^{2d+2\ell}, \ldots, X^{3d+2\ell-1}$.
- For lower queries, we simulate $P_{d+1}, \ldots, P_{2d+2\ell-1}$ through lazy-sampling to define $X^{d+1}, \ldots, X^{3d+2\ell-1}$, and we use the inverse of the $dn$-bit random permutation $\pi^{-1}$ to define $X^1, \ldots, X^d$.

Recall that the simulation is for primitive queries in the ideal world, and upper queries may include both queries to request encryption $P_x(\cdot, \cdot)$ and decryption $P_x^{-1}(\cdot, \cdot)$ for $x$ in a certain range. Nevertheless, for all these queries, we complete the computation of the internal variables by using $\pi$. Similarly, lower queries may include both queries to request encryption and decryption of the primitive. For lower queries, we use $\pi^{-1}$ to complete the computation of the internal variables, i.e., the use of $\pi/\pi^{-1}$ can be viewed as the distinction between upper/lower queries.

With all these definitions of the oracles, we can define transcripts that summarize the interaction between the adversary and the oracles. In order to use the Coefficient-H technique in Lemma 1, we define good and bad transcripts, where the good transcripts are those that can occur with a non-zero probability in the real world, and the bad transcripts are the complement of the good transcripts.

The proof now reduces to bound $\epsilon_1$ and $\epsilon_2$ in Lemma 1, i.e., the probability that we have a bad transcript in the ideal world (Lemma 2) and the ratio of the interpolation probabilities (Lemma 3).

The probability that we have a bad transcript is reduced to analyze the probability of certain bad events in the ideal world, and this can be proved to be sufficiently small as all the events are defined so that they involve collisions between $(\ell+1)n$-bit random variables. In the course of deriving the probability, we rely on the assumption of $q \leq 2^n$. Our proof strategy of giving all the internal variables to the adversary makes it possible to define bad events with $(\ell+1)n$-bit random variables, while at the same time it introduces the assumption of $q \leq 2^n$ to obtain the final bound.

The analysis of the ratio between the interpolation probabilities is relatively simple, and we complete the proof of Theorem 1 from Lemma 1.

# 5 Proof of Theorem 1

In this section, we present our proof of Theorem 1.

We start with defining the oracles $\Phi$, $P$, $\Pi$, and $S$ of Definition 1. $\Phi$ represents $\Phi_{2d+2\ell-1}$, $P$ represents $P_1, \ldots, P_{2d+2\ell-1}$, $\Pi$ represents a $dn$-bit random permutation $\pi \xleftarrow{\$} \mathrm{Perm}(dn)$, and $S$ represents a simulator.

A query to the construction oracle $\mathcal{O}_c \in \{\Phi, \Pi\}$ is denoted as $(\delta, \bar{X})$, where $\delta \in \{+, -\}$ represents the query direction and $\bar{X} \in \{0,1\}^{dn}$ is the input, i.e., for a query $(+, \bar{X})$, the oracle returns $\Phi(\bar{X})$ or $\Pi(\bar{X})$, and for a query $(-, \bar{X})$, the oracle returns $\Phi^{-1}(\bar{X})$ or $\Pi^{-1}(\bar{X})$. A query to the primitive oracle $\mathcal{O}_p \in \{P, S\}$ is denoted as $(\delta, x, \bar{X})$, where $x \in [1..2d+2\ell-1]$ denotes the index of the underlying ideal cipher, i.e., when $\mathcal{O}_p = P$, the oracle returns $P_x(X^{[2..d]}, X^1)$ for a query $(+, x, \bar{X})$, where $\bar{X} = X^{[1..d]}$, and for a query

$(-, x, \bar{X})$, the oracle returns $P_x^{-1}(X^{[1..d-1]}, X^d)$, where $\bar{X} = X^{[1..d]}$. When $\mathcal{O}_p = S$, we define the behavior of the simulator below.

In what follows, we assume that $\mathcal{A}$ makes exactly $q_c$ queries to $\mathcal{O}_c$ and $q_c + q_p$ queries to $\mathcal{O}_p$, and if $\mathcal{A}$ makes a query to $\mathcal{O}_c$, then we force $\mathcal{A}$ to make another query to $\mathcal{O}_p$ immediately. Specifically, if $\mathcal{A}$ makes a construction query $(+, \bar{X})$ (resp. $(-, \bar{X})$) to $\mathcal{O}_c$, then we force $\mathcal{A}$ to immediately make a primitive query $(+, 1, \bar{X})$ (resp. $(-, 2d+2\ell-1, \bar{X})$) to $\mathcal{O}_p$ after the construction query. We see that from any adversary that makes at most $q_c$ queries to $\mathcal{O}_c$ and at most $q_p$ queries to $\mathcal{O}_p$, we can build $\mathcal{A}$ with the same output that satisfies this property, and we show that (1) holds for this adversary that makes extra primitive queries. We remark that the new adversary is more powerful in that it receives more information than the original adversary. Our approach is to show that the adversary has a low distinguishing advantage, even if it receives extra information from the oracles.

**Real World Oracles.** Now in the real world, i.e., when $(\mathcal{O}_c, \mathcal{O}_p) = (\Phi, P)$, we define $\Phi$ as in Algorithms 1 and 2 in Fig. 4, and $P$ as in Algorithms 3 and 4 in Fig. 5.

The procedure of $P$ is defined so that it returns all the internal variables, and in what follows, $P$ represents the oracle that returns the extra information. For the $i$-th primitive query $(\delta_i, x_i, \bar{X}_i)$, where $\delta_i = +$ and $\bar{X}_i = X_i^{[x_i..x_i+d-1]}$, $P$ computes $X_i^{x_i+d} \leftarrow P_{x_i}(X_i^{[x_i+1..x_i+d-1]}, X_i^{x_i})$ and returns it to $\mathcal{A}$ or $\Phi$. $P$ also computes $X_i^{[1..x_i-1]}$ and $X_i^{[x_i+d+1..3d+2\ell-1]}$ by following the definition of $\Phi_{2d+2\ell-1}$ and returns all the values; $P$ completes $X_i^1, \ldots, X_i^{3d+2\ell-1}$ that form all the internal variables of $\Phi_{2d+2\ell-1}$ from $\bar{X}_i = X_i^{[x_i..x_i+d-1]}$ by using $P_1^{-1}, \ldots, P_{x_i-1}^{-1}$ and $P_{x_i}, \ldots, P_{2d+2\ell-1}$. The behavior of $P$ for the $i$-th query $(\delta_i, x_i, \bar{X}_i)$, where $\delta_i = -$, is similarly defined. $\Phi$ is defined by using the output of $P$ that contains the output of $\Phi_{2d+2\ell-1}$.

**Example 1.** For example, let $d = 3$ and $\ell = 1$, and consider $\Phi_7$ in Fig. 6. If $\mathcal{A}$'s query to $P$ is $(+, 4, X^{[4..6]})$, then, $P$ computes

$$
\begin{aligned}
X^3 &\leftarrow P_3^{-1}(X^4 \,\|\, X^5, X^6) \\
X^2 &\leftarrow P_2^{-1}(X^3 \,\|\, X^4, X^5) \\
X^1 &\leftarrow P_1^{-1}(X^2 \,\|\, X^3, X^4)
\end{aligned}
$$

and

$$
\begin{aligned}
X^7 &\leftarrow P_4(X^5 \,\|\, X^6, X^4) \\
X^8 &\leftarrow P_5(X^6 \,\|\, X^7, X^5) \\
X^9 &\leftarrow P_6(X^7 \,\|\, X^8, X^6) \\
X^{10} &\leftarrow P_7(X^8 \,\|\, X^9, X^7)\,,
\end{aligned}
$$

and returns $X^{[1..3]} \,\|\, X^{[7..10]}$ to $\mathcal{A}$. For $\mathcal{A}$'s query $(-, 3, X^{[4..6]})$ to $P$, $P$ executes the same computation and returns $X^{[1..3]} \,\|\, X^{[7..10]}$ to $\mathcal{A}$. We also consider $\mathcal{A}$'s query $(+, X^{[1..3]})$ to $\Phi$. Then, $\Phi$ makes a query $(+, 1, X^{[1..3]})$ to $P$ and receives $X^{[4..10]}$. It returns $X^{[8..10]}$ to $\mathcal{A}$, which is the last $3n$-bit string of the response $X^{[4..10]}$ from $P$.

We note that since the adversary is forced to make a primitive query and receives all the internal variables, a construction query may seem to be redundant. However, we still require the adversary to make construction queries so that in the ideal world, the simulation of the simulator will not be easier.

**Algorithm 1:** Procedure of $\Phi$ for the $i$-th query $(\delta_i, \bar{X}_i)$ with $\delta_i = +$

**Input:** $\bar{X}_i = X_i^{[1..d]} \in \{0,1\}^{dn}$
**Output:** $X_i^{[2d+2\ell..3d+2\ell-1]} \in \{0,1\}^{dn}$

1. $X_i^{[d+1..3d+2\ell-1]} \leftarrow P(+, 1, X_i^{[1..d]})$
2. **return** $X_i^{[2d+2\ell..3d+2\ell-1]}$

**Algorithm 2:** Procedure of $\Phi$ for the $i$-th query $(\delta_i, \bar{X}_i)$ with $\delta_i = -$

**Input:** $\bar{X}_i = X_i^{[2d+2\ell..3d+2\ell-1]} \in \{0,1\}^{dn}$
**Output:** $X_i^{[1..d]} \in \{0,1\}^{dn}$

1. $X_i^{[1..2d+2\ell-1]} \leftarrow P(-, 2d+2\ell-1, X_i^{[2d+2\ell..3d+2\ell-1]})$
2. **return** $X_i^{[1..d]}$

**Figure 4:** Procedure of $\Phi$. It internally invokes $P(+, 1, \cdot)$ or $P(-, 2d+2\ell-1, \cdot)$, and returns a part of their output.

**Algorithm 3:** Procedure of $P$ for the $i$-th query $(\delta_i, x_i, \bar{X}_i)$ with $\delta_i = +$

**Input:** $x_i \in [1..2d+2\ell-1]$, $\bar{X}_i = X_i^{[x_i..x_i+d-1]} \in \{0,1\}^{dn}$
**Output:** $X_i^{[1..x_i-1]} \,\|\, X_i^{[x_i+d..3d+2\ell-1]} \in \{0,1\}^{(2d+2\ell-1)n}$

1. **for** $y = x_i - 1, \ldots, 1$ **do** $(x_i \neq 1)$
   $X_i^y \leftarrow P_y^{-1}(X_i^{[y+1..y+d-1]}, X_i^{y+d})$
2. **for** $y = x_i, \ldots, 2d+2\ell-1$ **do**
   $X_i^{y+d} \leftarrow P_y(X_i^{[y+1..y+d-1]}, X_i^y)$
3. **return** $X_i^{[1..x_i-1]} \,\|\, X_i^{[x_i+d..3d+2\ell-1]}$

**Algorithm 4:** Procedure of $P$ for the $i$-th query $(\delta_i, x_i, \bar{X}_i)$ with $\delta_i = -$

**Input:** $x_i \in [1..2d+2\ell-1]$, $\bar{X}_i = X_i^{[x_i+1..x_i+d]} \in \{0,1\}^{dn}$
**Output:** $X_i^{[1..x_i]} \,\|\, X_i^{[x_i+d+1..3d+2\ell-1]} \in \{0,1\}^{(2d+2\ell-1)n}$

1. **for** $y = x_i, \ldots, 1$ **do**
   $X_i^y \leftarrow P_y^{-1}(X_i^{[y+1..y+d-1]}, X_i^{y+d})$
2. **for** $y = x_i + 1, \ldots, 2d+2\ell-1$ **do** $(x_i \neq 2d+2\ell-1)$
   $X_i^{y+d} \leftarrow P_y(X_i^{[y+1..y+d-1]}, X_i^y)$
3. **return** $X_i^{[1..x_i]} \,\|\, X_i^{[x_i+d+1..3d+2\ell-1]}$

**Figure 5:** Procedure of $P$. If $\delta_i = +$, then it internally invokes $P_1^{-1}, \ldots, P_{x_i-1}^{-1}$ and $P_{x_i}, \ldots, P_{2d+2\ell-1}$, and returns all the internal variables. The case $\delta_i = -$ is analogously defined.

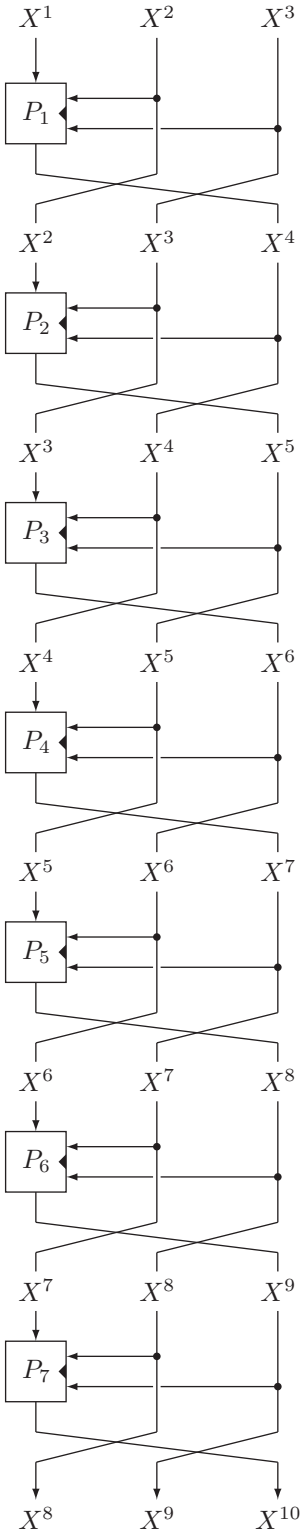



**Figure 6:** $\Phi_7[E_1, \ldots, E_7]$ for $d = 3$ and $\ell = 1$

**Ideal World Oracles.** Next, in the ideal world, i.e., when $(\mathcal{O}_c, \mathcal{O}_p) = (\Pi, S)$, we define $\Pi$ as in Algorithms 5 and 6 in Fig. 7 and $S$ as in Algorithms 7 and 8 in Fig. 8.

The definition of $\Pi$ in Algorithms 5 and 6 is straightforward. The definition of $S$ in Algorithms 7 and 8 simulates $P_1, \ldots, P_{2d+2\ell+1}$ (and their decryption functions) with the lazy-sampling approach. In Fig. 8, for $y \in [1..2d + 2\ell - 1]$, the computation

$$\mathcal{X}_i^y \leftarrow \{X_j^{y+d} \mid j < i \wedge X_i^{[y+1..y+d-1]} = X_j^{[y+1..y+d-1]}\}$$
$$X_i^{y+d} \xleftarrow{\$} \{0,1\}^n \setminus \mathcal{X}_i^y$$

simulates $X_i^{y+d} \leftarrow P_y(X_i^{[y+1..y+d-1]}, X_i^y)$, and the computation

$$\mathcal{X}_i^y \leftarrow \{X_j^y \mid j < i \wedge X_i^{[y+1..y+d-1]} = X_j^{[y+1..y+d-1]}\}$$
$$X_i^y \xleftarrow{\$} \{0,1\}^n \setminus \mathcal{X}_i^y$$

simulates $X_i^y \leftarrow P_y^{-1}(X_i^{[y+1..y+d-1]}, X_i^{y+d})$. Observe that it works as follows:

- If the $i$-th primitive query is $(\delta_i, x_i, \bar{X}_i)$ with $x_i \in [1..d+\ell-1]$ or $(\delta_i, x_i, \bar{X}_i)$ with $\delta_i = +$ and $x_i = d + \ell$, then $S$ simulates $P_1, \ldots, P_{d+\ell-1}$ (or their decryption functions) and $P_{d+\ell}, \ldots, P_{d+2\ell-1}$, and computes $X_i^1, \ldots, X_i^{3d+2\ell-1}$, where $X_i^1, \ldots, X_i^{2d+2\ell-1}$ are inputs or computed as in the real world, and the remaining values $X_i^{2d+2\ell}, \ldots, X_i^{3d+2\ell-1}$ are computed with $\pi$.
- On the other hand, if the $i$-th primitive query is $(\delta_i, x_i, \bar{X}_i)$ with $x_i \in [d + \ell + 1..2d + 2\ell - 1]$ or $(\delta_i, x_i, \bar{X}_i)$ with $\delta_i = -$ and $x_i = d + \ell$, then $S$ simulates $P_{d+1}^{-1}, \ldots, P_{d+\ell}^{-1}$ and $P_{d+\ell+1}, \ldots, P_{2d+2\ell-1}$ (or their decryption functions), and computes $X_i^1, \ldots, X_i^{3d+2\ell-1}$, where $X_i^{d+1}, \ldots, X_i^{3d+2\ell-1}$ are inputs or computed as in the real world, and the remaining values $X_i^1, \ldots, X_i^d$ are computed with $\pi^{-1}$.

The above observation motivates us to define an *upper query* and a *lower query*.

We call a primitive query $(\delta, x, \bar{X})$ an upper query if $x \in [1..d+\ell-1]$ or $\delta = + \wedge x = d+\ell$, and we call a query that is not an upper query a lower query; a query $(\delta, x, \bar{X})$ is a lower query if $x \in [d + \ell + 1..2d + 2\ell - 1]$ or $\delta = - \wedge x = d + \ell$.

**Example 2.** For example, let $d = 3$ and $\ell = 1$, and consider $\Phi_7$ in Fig. 6. Then, a query $(\delta, x, \bar{X})$ to $S$, where $\bar{X} \in \{X^{[1..3]}, X^{[2..4]}, X^{[3..5]}, X^{[4..6]}\}$, is an upper query regardless of the query direction $\delta$. Similarly, a query $(\delta, x, \bar{X})$ to $S$, where $\bar{X} \in \{X^{[5..7]}, X^{[6..8]}, X^{[7..9]}, X^{[8..10]}\}$, is a lower query regardless of $\delta$.

For an upper query, $S$ simulates $P_1$ or $P_1^{-1}$, $P_2$ or $P_2^{-1}$, $P_3$ or $P_3^{-1}$, and $P_4$, and for a lower query, it simulates $P_5$ or $P_5^{-1}$, $P_6$ or $P_6^{-1}$, $P_7$ or $P_7^{-1}$, and $P_4^{-1}$. For example, for $\mathcal{A}$'s upper query $(+, 4, X^{[4..6]})$ to $S$, $S$ simulates $P_1^{-1}$, $P_2^{-1}$, $P_3^{-1}$, and $P_4$, and computes

$$X^3 \leftarrow P_3^{-1}(X^4 \,\|\, X^5, X^6)$$
$$X^2 \leftarrow P_2^{-1}(X^3 \,\|\, X^4, X^5)$$
$$X^1 \leftarrow P_1^{-1}(X^2 \,\|\, X^3, X^4)$$

and

$$X^7 \leftarrow P_4(X^5 \,\|\, X^6, X^4)\,.$$

Then, it makes a query $(+, X^{[1..3]})$ to $\Pi$ and receives $X^{[8..10]}$. It returns $X^{[1..3]} \,\|\, X^{[7..10]}$ to $\mathcal{A}$.

**Algorithm 5:** Procedure of $\Pi$ for the $i$-th query $(\delta_i, \bar{X}_i)$ with $\delta_i = +$

**Input:** $\bar{X}_i = X_i^{[1..d]} \in \{0,1\}^{dn}$
**Output:** $X_i^{[2d+2\ell..3d+2\ell-1]} \in \{0,1\}^{dn}$

1. $X_i^{[2d+2\ell..3d+2\ell-1]} \leftarrow \pi(X_i^{[1..d]})$
2. **return** $X_i^{[2d+2\ell..3d+2\ell-1]}$

**Algorithm 6:** Procedure of $\Pi$ for the $i$-th query $(\delta_i, \bar{X}_i)$ with $\delta_i = -$

**Input:** $\bar{X}_i = X_i^{[2d+2\ell..3d+2\ell-1]} \in \{0,1\}^{dn}$
**Output:** $X_i^{[1..d]} \in \{0,1\}^{dn}$

1. $X_i^{[1..d]} \leftarrow \pi^{-1}(X_i^{[2d+2\ell..3d+2\ell-1]})$
2. **return** $X_i^{[1..d]}$

**Figure 7:** Procedure of $\Pi$. The definition is straightforward.

**Transcripts.** We assume without loss of generality that $\mathcal{A}$ is deterministic, does not repeat a query, and does not make a redundant query. This implies that:

- If $\mathcal{A}$ makes a query $(+, X^{[1..d]})$ to $\mathcal{O}_c$ and obtains $X^{[2d+2\ell..3d+2\ell-1]}$, then it does not make a query $(-, X^{[2d+2\ell..3d+2\ell-1]})$ to $\mathcal{O}_c$, and vice versa.
- If $\mathcal{A}$ makes a query $(+, x, X^{[x..x+d-1]})$ to $\mathcal{O}_p$ and obtains $X^{[1..x-1]} \,\|\, X^{[x+d..3d+2\ell-1]}$ or it makes a query $(-, x, X^{[x+1..x+d]})$ to $\mathcal{O}_p$ and obtains $X^{[1..x]} \,\|\, X^{[x+d+1..3d+2\ell-1]}$, then it does not make a query $(+, X^{[1..d]})$ nor $(-, X^{[2d+2\ell..3d+2\ell-1]})$ to $\mathcal{O}_c$, and it does not make a query $(+, y, X^{[y..y+d-1]})$ nor $(-, y, X^{[y+1..y+d]})$ for any $y \in [1..2d+2\ell-1]$ to $\mathcal{O}_p$.

Since $\mathcal{A}$ is deterministic and makes $q_c$ queries to $\mathcal{O}_c$ and $q_c + q_p$ queries to $\mathcal{O}_p$, these queries and responses can be summarized in a transcript $\tau = (\tau_c, \tau_p)$, where

$$\tau_c = ((\delta_1, X_1^{[1..d]}, X_1^{[2d+2\ell..3d+2\ell-1]}), \ldots, (\delta_{q_c}, X_{q_c}^{[1..d]}, X_{q_c}^{[2d+2\ell..3d+2\ell-1]}))$$

and

$$\tau_p = ((\delta_1, x_1, X_1^{[1..3d+2\ell-1]}), \ldots, (\delta_{q_c+q_p}, x_{q_c+q_p}, X_{q_c+q_p}^{[1..3d+2\ell-1]}))\,.$$

We denote by $T^{\mathrm{re}}$ (resp. $T^{\mathrm{id}}$) the probability distribution of transcripts when $\mathcal{A}$ interacts with $(\Phi, P)$ (resp. $(\Pi, S)$).

We consider an *attainable transcript* $\tau = (\tau_c, \tau_p)$; $\tau_c$ that can be obtained with interacting with $\Pi$ and $\tau_p$ that can be obtained with interacting $S$. Since $\mathcal{A}$ does not repeat a query and does not make a redundant query, it holds in $\tau_p$ that

$$X_i^{[1..3d+2\ell-1]} \neq X_j^{[1..3d+2\ell-1]}$$

for any $1 \le j < i \le q_c + q_p$. In the real world, we also see that

$$X_i^{[x..x+d-1]} \neq X_j^{[x..x+d-1]} \tag{2}$$

holds for any $x \in [1..2d + 2\ell]$ and $1 \le j < i \le q_c + q_p$, since each of the round $\varphi[P_1], \ldots, \varphi[P_{2d+2\ell-1}]$ is a permutation on $\{0,1\}^{dn}$. From the same reasoning, in the ideal world, it holds that

$$X_i^{[x..x+d-1]} \neq X_j^{[x..x+d-1]} \tag{3}$$

**Algorithm 7:** Procedure of $S$ for the $i$-th query $(\delta_i, x_i, \bar{X}_i)$ with $\delta_i = +$

**Input:** $x_i \in [1..2d+2\ell-1]$, $\bar{X}_i = X_i^{[x_i..x_i+d-1]} \in \{0,1\}^{dn}$
**Output:** $X_i^{[1..x_i-1]} \,\|\, X_i^{[x_i+d..3d+2\ell-1]} \in \{0,1\}^{(2d+2\ell-1)n}$

1. **if** $x_i \in [1..d+\ell]$
   **for** $y = x_i - 1, \ldots, 1$ **do** $(x_i \neq 1)$
   $\mathcal{X}_i^y \leftarrow \{X_j^y \mid j < i \wedge X_i^{[y+1..y+d-1]} = X_j^{[y+1..y+d-1]}\}$
   $X_i^y \xleftarrow{\$} \{0,1\}^n \setminus \mathcal{X}_i^y$
   **for** $y = x_i, \ldots, d+2\ell-1$ **do**
   $\mathcal{X}_i^y \leftarrow \{X_j^{y+d} \mid j < i \wedge X_i^{[y+1..y+d-1]} = X_j^{[y+1..y+d-1]}\}$
   $X_i^{y+d} \xleftarrow{\$} \{0,1\}^n \setminus \mathcal{X}_i^y$
   $X_i^{[2d+2\ell..3d+2\ell-1]} \leftarrow \Pi(+, X_i^{[1..d]})$
2. **else**
   **for** $y = x_i - 1, \ldots, d+1$ **do**
   $\mathcal{X}_i^y \leftarrow \{X_j^y \mid j < i \wedge X_i^{[y+1..y+d-1]} = X_j^{[y+1..y+d-1]}\}$
   $X_i^y \xleftarrow{\$} \{0,1\}^n \setminus \mathcal{X}_i^y$
   **for** $y = x_i, \ldots, 2d+2\ell-1$ **do**
   $\mathcal{X}_i^y \leftarrow \{X_j^{y+d} \mid j < i \wedge X_i^{[y+1..y+d-1]} = X_j^{[y+1..y+d-1]}\}$
   $X_i^{y+d} \xleftarrow{\$} \{0,1\}^n \setminus \mathcal{X}_i^y$
   $X_i^{[1..d]} \leftarrow \Pi(-, X_i^{[2d+2\ell..3d+2\ell-1]})$
3. **return** $X_i^{[1..x_i-1]} \,\|\, X_i^{[x_i+d..3d+2\ell-1]}$

**Algorithm 8:** Procedure of $S$ for the $i$-th query $(\delta_i, x_i, \bar{X}_i)$ with $\delta_i = -$

**Input:** $x_i \in [1..2d+2\ell-1]$, $\bar{X}_i = X_i^{[x_i+1..x_i+d]} \in \{0,1\}^{dn}$
**Output:** $X_i^{[1..x_i]} \,\|\, X_i^{[x_i+d+1..3d+2\ell-1]} \in \{0,1\}^{(2d+2\ell-1)n}$

1. **if** $x_i \in [1..d+\ell-1]$
   **for** $y = x_i, \ldots, 1$ **do**
   $\mathcal{X}_i^y \leftarrow \{X_j^y \mid j < i \wedge X_i^{[y+1..y+d-1]} = X_j^{[y+1..y+d-1]}\}$
   $X_i^y \xleftarrow{\$} \{0,1\}^n \setminus \mathcal{X}_i^y$
   **for** $y = x_i + 1, \ldots, d+2\ell-1$ **do**
   $\mathcal{X}_i^y \leftarrow \{X_j^{y+d} \mid j < i \wedge X_i^{[y+1..y+d-1]} = X_j^{[y+1..y+d-1]}\}$
   $X_i^{y+d} \xleftarrow{\$} \{0,1\}^n \setminus \mathcal{X}_i^y$
   $X_i^{[2d+2\ell..3d+2\ell-1]} \leftarrow \Pi(+, X_i^{[1..d]})$
2. **else**
   **for** $y = x_i, \ldots, d+1$ **do**
   $\mathcal{X}_i^y \leftarrow \{X_j^y \mid j < i \wedge X_i^{[y+1..y+d-1]} = X_j^{[y+1..y+d-1]}\}$
   $X_i^y \xleftarrow{\$} \{0,1\}^n \setminus \mathcal{X}_i^y$
   **for** $y = x_i + 1, \ldots, 2d+2\ell-1$ **do** $(x_i \neq 2d+2\ell-1)$
   $\mathcal{X}_i^y \leftarrow \{X_j^{y+d} \mid j < i \wedge X_i^{[y+1..y+d-1]} = X_j^{[y+1..y+d-1]}\}$
   $X_i^{y+d} \xleftarrow{\$} \{0,1\}^n \setminus \mathcal{X}_i^y$
   $X_i^{[1..d]} \leftarrow \Pi(-, X_i^{[2d+2\ell..3d+2\ell-1]})$
3. **return** $X_i^{[1..x_i]} \,\|\, X_i^{[x_i+d+1..3d+2\ell-1]}$

**Figure 8:** Procedure of $S$. See the main body for the explanation. Note that line 1 in both Algorithms 7 and 8 handles upper queries, and they are essentially the same. Similarly, line 2 in both Algorithms 7 and 8 handles lower queries.

for any $x \in [1..d+2\ell] \cup \{2d+2\ell\}$ and $1 \le j < i \le q_c + q_p$ when the $i$-th query is an upper query, and for any $x \in [d+1..2d+2\ell] \cup \{1\}$ and $1 \le j < i \le q_c + q_p$ when the $i$-th query is a lower query.

**Example 3.** For example, let $d = 3$ and $\ell = 1$, and consider $\Phi_7$ in Fig. 6. In the real world, (2) states that the following inequalities hold for any $1 \le j < i \le q_c + q_p$:

$$\begin{aligned}
X_i^{[1..3]} &\neq X_j^{[1..3]} \\
X_i^{[2..4]} &\neq X_j^{[2..4]} \\
X_i^{[3..5]} &\neq X_j^{[3..5]} \\
X_i^{[4..6]} &\neq X_j^{[4..6]} \\
X_i^{[5..7]} &\neq X_j^{[5..7]} \\
X_i^{[6..8]} &\neq X_j^{[6..8]} \\
X_i^{[7..9]} &\neq X_j^{[7..9]} \\
X_i^{[8..10]} &\neq X_j^{[8..10]}
\end{aligned}$$

In the ideal world, on the other hand, (3) states that the inequalities $X_i^{[1..3]} \neq X_j^{[1..3]}$, $X_i^{[2..4]} \neq X_j^{[2..4]}$, $X_i^{[3..5]} \neq X_j^{[3..5]}$, $X_i^{[4..6]} \neq X_j^{[4..6]}$, $X_i^{[5..7]} \neq X_j^{[5..7]}$, and $X_i^{[8..10]} \neq X_j^{[8..10]}$ hold for any $1 \le j < i \le q_c + q_p$ if the $i$-th query is an upper query. However, there is no guarantee that we have $X_i^{[6..8]} \neq X_j^{[6..8]}$ or $X_i^{[7..9]} \neq X_j^{[7..9]}$. Similarly, if the $i$-th query is a lower query, then we have $X_i^{[1..3]} \neq X_j^{[1..3]}$, $X_i^{[4..6]} \neq X_j^{[4..6]}$, $X_i^{[5..7]} \neq X_j^{[5..7]}$, $X_i^{[6..8]} \neq X_j^{[6..8]}$, $X_i^{[7..9]} \neq X_j^{[7..9]}$, and $X_i^{[8..10]} \neq X_j^{[8..10]}$. However, $X_i^{[2..4]} = X_j^{[2..4]}$ or $X_i^{[3..5]} = X_j^{[3..5]}$ may hold, which follows from the definition of the simulator.

We now define the set of good transcripts as

$$\mathcal{T}^{\mathrm{good}} = \{\tau = (\tau_c, \tau_p) \mid \Pr[T^{\mathrm{re}} = \tau] > 0\}.$$

The set of bad transcripts is defined as $\mathcal{T}^{\mathrm{bad}} = \mathcal{T}^{\mathrm{all}} \setminus \mathcal{T}^{\mathrm{good}}$, where $\mathcal{T}^{\mathrm{all}}$ is the set of all attainable transcripts. Recall that we call a transcript $\tau$ attainable if $\Pr[T^{\mathrm{id}} = \tau] > 0$ holds, i.e., if $\tau$ can be obtained with interacting $(\Pi, S)$.

**Evaluation of $\Pr[T^{\mathrm{id}} \in \mathcal{T}^{\mathrm{bad}}]$.** We consider $\tau \in \mathcal{T}^{\mathrm{bad}}$ first. We prove the following lemma.

**Lemma 2.** *Let $q_c + q_p \le 2^n$. Then* $\Pr[T^{\mathrm{id}} \in \mathcal{T}^{\mathrm{bad}}] \le \dfrac{(q_c+q_p)^2}{2^{(\ell+1)n}} + \dfrac{(\ell-1)(q_c+q_p)^2}{2^{dn}}$.

*Proof.* In the ideal world, from the definition of $\mathcal{T}^{\mathrm{bad}}$, we have $(\tau_c, \tau_p) \in \mathcal{T}^{\mathrm{bad}}$ if it holds in $\tau_p$ that

$$X_i^{[x..x+d-1]} = X_j^{[x..x+d-1]}$$

for some $x \in [2..2d+2\ell-1]$ and $1 \le j < i \le q_c + q_p$.

**Example 4.** For example, let $d = 3$ and $\ell = 1$. We have $(\tau_c, \tau_p) \in \mathcal{T}^{\mathrm{bad}}$ if $X_i^{[2..4]} = X_j^{[2..4]}$, $X_i^{[3..5]} = X_j^{[3..5]}$, $X_i^{[4..6]} = X_j^{[4..6]}$, $X_i^{[5..7]} = X_j^{[5..7]}$, $X_i^{[6..8]} = X_j^{[6..8]}$, or $X_i^{[7..9]} = X_j^{[7..9]}$ holds for some $1 \le j < i \le q_c + q_p$.[2]

[2] We note that, even though they are included for notational simplicity, $X_i^{[4..6]} = X_j^{[4..6]}$ and $X_i^{[5..7]} = X_j^{[5..7]}$ can never happen. We also note that $X_i^{[1..3]} = X_j^{[1..3]}$ and $X_i^{[8..10]} = X_j^{[8..10]}$ cannot happen as the adversary does not repeat a query and does not make a redundant query.

We denote by $i_s$ the smallest $i$ such that $X_i^{[x..x+d-1]} = X_j^{[x..x+d-1]}$ holds for some $x \in [2..2d+2\ell-1]$ and $j \in [1..i-1]$ in $\tau_p$. We have

$$\Pr[T^{\mathrm{id}} \in \mathcal{T}^{\mathrm{bad}}] \leq \sum_{i \in [1..q_c+q_p]} \Pr[i_s = i]$$

and $\Pr[i_s = i] = 0$ for $i = 1$.

We next analyze $\Pr[i_s = i]$ when the $i$-th query is a lower query. Then, $\Pr[i_s = i]$ is the probability that at least one of the following collisions occurs for some $j \in [1..i-1]$ in the ideal world:

$$\begin{aligned} X_i^{[1..d]} &= X_j^{[1..d]} \\ &\vdots \\ X_i^{[\ell..d+\ell-1]} &= X_j^{[\ell..d+\ell-1]} \\ X_i^{[\ell+1..d-1]} \,\|\, X_i^{[d..d+\ell]} &= X_j^{[\ell+1..d-1]} \,\|\, X_j^{[d..d+\ell]} \\ &\vdots \\ X_i^{[d..d+\ell]} \,\|\, X_i^{[d+\ell+1..2d-1]} &= X_j^{[d..d+\ell]} \,\|\, X_j^{[d+\ell+1..2d-1]} \end{aligned} \tag{4}$$

Here, $\Pr[X_i^{[1..d]} = X_j^{[1..d]}] = 0$ since $\mathcal{A}$ does not repeat a query and does not make a redundant query. Note that $i_s$ is the smallest index that we have a collision, and thus when we consider $\Pr[i_s = i]$, we assume that, for $x \in [2..2d+2\ell-1]$, there does not exist a collision between the elements in $\{X_j^{[x..x+d-1]} \mid j \in [1..i-1]\}$.

**Example 5.** For example, let $d = 3$ and $\ell = 1$. Then (4) states that $\Pr[i_s = i]$ is the probability of $X_i^{[1..3]} = X_j^{[1..3]}$, $X_i^2 \,\|\, X_i^{[3..4]} = X_j^2 \,\|\, X_j^{[3..4]}$, or $X_i^{[3..4]} \,\|\, X_i^5 = X_j^{[3..4]} \,\|\, X_j^5$ holds for some $j \in [1..i-1]$. Among them, $X_i^{[1..3]} = X_j^{[1..3]}$ does not occur. Here, we assume that $\{X_j^{[2..4]} \mid j \in [1..i-1]\}$, $\{X_j^{[3..5]} \mid j \in [1..i-1]\}$, $\{X_j^{[4..6]} \mid j \in [1..i-1]\}$, $\{X_j^{[5..7]} \mid j \in [1..i-1]\}$, $\{X_j^{[6..8]} \mid j \in [1..i-1]\}$, and $\{X_j^{[7..9]} \mid j \in [1..i-1]\}$ do not contain a collision.

Then, we have

$$\Pr[i_s = i] \leq \sum_{j \in [1..i-1]} \Pr[X_i^{[d..d+\ell]} = X_j^{[d..d+\ell]}] \tag{5}$$

$$+ \sum_{x \in [2..\ell]} \sum_{j \in [1..i-1]} \Pr[X_i^{[x..x+d-1]} = X_j^{[x..x+d-1]}] \tag{6}$$

when $\ell \in [2..d-1]$, and

$$\Pr[i_s = i] \leq \sum_{j \in [1..i-1]} \Pr[X_i^{[d..d+1]} = X_j^{[d..d+1]}] \tag{7}$$

when $\ell = 1$. Here, the first $\ell$ collisions in (4) appear in (6), and the last $d - \ell$ collisions in (4) appear in (5), where in the latter case, we focus on $X_i^{[d..d+\ell]}$ and ignore the rest. When $\ell = 1$, we focus on the last $\ell = 1$ collision in (4) to obtain (7).

Now, $X_1^{[2..d+\ell]}, \ldots, X_{i-1}^{[2..d+\ell]}$ are given to $\mathcal{A}$ and thus are fixed strings, and $X_i^2, \ldots, X_i^d$ are the random variables generated by the $dn$-bit random permutation. Note that, for the $i$-th query to $S$, $X_i^{[1..d]}$ generated by the $dn$-bit random permutation is selected uniformly at random from the set of size $2^{dn} - (i-1)$ when the $i$-th query is a lower query, because of the assumption of $\mathcal{A}$'s order of queries. Therefore, we have for $\ell \in [2..d-1]$,

$$\Pr[i_s = i] \leq \left( \sum_{j \in [1..i-1]} \Pr[X_i^{[d+1..d+\ell]} = X_j^{[d+1..d+\ell]}] \right) \cdot \frac{2^{(d-1)n}}{2^{dn} - (i-1)} \tag{8}$$

$$+ \sum_{x\in[2..\ell]} \left( \left( \sum_{j\in[1..i-1]} \Pr[X_i^{[d+1..x+d-1]} = X_j^{[d+1..x+d-1]}] \right) \cdot \frac{2^{(x-1)n}}{2^{dn}-(i-1)} \right), \tag{9}$$

where (8) is obtained by extracting $X_i^d$ from (5) that appears as $2^{(d-1)n}/(2^{dn}-(i-1))$, and (9) is obtained by extracting $X_i^{[x..d]}$ from (6) that appears as $2^{(x-1)n}/(2^{dn}-(i-1))$. We make a minor adjustment to the range of $x$ in (9) to obtain

$$\Pr[i_s = i] \le \left( \sum_{j\in[1..i-1]} \Pr[X_i^{[d+1..d+\ell]} = X_j^{[d+1..d+\ell]}] \right) \cdot \frac{2^{(d-1)n}}{2^{dn}-(i-1)} + \sum_{x\in[1..\ell-1]} \left( \left( \sum_{j\in[1..i-1]} \Pr[X_i^{[d+1..d+x]} = X_j^{[d+1..d+x]}] \right) \cdot \frac{2^{xn}}{2^{dn}-(i-1)} \right).$$

From Algorithms 7 and 8, for $y = \ell, \ldots, 1$, we let $X_i^{d+y} \xleftarrow{\$} \{0,1\}^n \setminus \mathcal{X}_i^{d+y}$. Therefore, for each $j \in [1..i-1]$ and $x \in [1..\ell]$, we have

$$\Pr[X_i^{[d+1..d+x]} = X_j^{[d+1..d+x]}] \le \frac{1}{\prod_{y\in[1..x]} (2^n - |\mathcal{X}_i^{d+y}|)}.$$

Our next task is to count the number of $j \in [1..i-1]$ satisfying $\Pr[X_i^{[d+1..d+x]} = X_j^{[d+1..d+x]}] = 0$. Assume that we have $X_j^{d+y} \in \mathcal{X}_i^{d+y}$ for some $y \in [1..x]$. Then, $X_i^{[d+y+1..2d+y-1]} = X_j^{[d+y+1..2d+y-1]}$ holds from the definition of $\mathcal{X}_i^{d+y}$, and $X_i^{d+y}$ is generated so that $X_i^{d+y} \neq X_j^{d+y}$ holds.

- From the former, it follows that $X_j^{d+z} \notin \mathcal{X}_i^{d+z}$ holds for all $z \in [y+1..x]$.
- From the latter, we have $\Pr[X_i^{[d+1..d+x]} = X_j^{[d+1..d+x]}] = 0$.

An important observation here is that if $X_j^{d+y} \in \mathcal{X}_i^{d+y}$ holds for some $y \in [1..x]$, then $X_j^{d+z} \notin \mathcal{X}_i^{d+z}$ holds for all $z \in [y+1..x]$. Since this holds true for all $y \in [1..x]$, the sets $\{j \mid X_j^{d+1} \in \mathcal{X}_i^{d+1}\}, \ldots, \{j \mid X_j^{d+x} \in \mathcal{X}_i^{d+x}\}$ do not contain common elements. Therefore, the number of $j \in [1..i-1]$ satisfying $\Pr[X_i^{[d+1..d+x]} = X_j^{[d+1..d+x]}] = 0$ is at least $\sum_{y\in[1..x]} |\{j \mid X_j^{d+y} \in \mathcal{X}_i^{d+y}\}| = \sum_{y\in[1..x]} |\mathcal{X}_i^{d+y}|$. From this observation, we obtain

$$\sum_{j\in[1..i-1]} \Pr[X_i^{[d+1..d+x]} = X_j^{[d+1..d+x]}] \le \frac{(i-1) - \left( \sum_{y\in[1..x]} |\mathcal{X}_i^{d+y}| \right)}{\prod_{y\in[1..x]} (2^n - |\mathcal{X}_i^{d+y}|)}.$$

At this point, we use the following inequality, which holds under the assumption of $q \le 2^n$.[3]

$$\frac{(i-1) - \left( \sum_{y\in[1..x]} |\mathcal{X}_i^{d+y}| \right)}{\prod_{y\in[1..x]} (2^n - |\mathcal{X}_i^{d+y}|)} \cdot \frac{1}{2^{dn}-(i-1)} \le \frac{2(i-1)}{2^{(d+x)n}}. \tag{10}$$

[3] We use the assumption of $q \le 2^n$ here.

The proof is elementary, and almost the same inequality was used in [NI19, Appendix A]. We present in Appendix A a proof for completeness.

Then, we have for $q \le 2^n$ and $\ell \in [2..d-1]$,

$$\begin{aligned}
\Pr[i_s = i] &\le \frac{(i-1) - \left(\sum_{y\in[1..\ell]} |\mathcal{X}_i^{d+y}|\right)}{\prod_{y\in[1..\ell]} (2^n - |\mathcal{X}_i^{d+y}|)} \cdot \frac{2^{(d-1)n}}{2^{dn} - (i-1)} \\
&\quad + \sum_{x\in[1..\ell-1]} \left( \frac{(i-1) - \left(\sum_{y\in[1..x]} |\mathcal{X}_i^{d+y}|\right)}{\prod_{y\in[1..x]} (2^n - |\mathcal{X}_i^{d+y}|)} \cdot \frac{2^{xn}}{2^{dn} - (i-1)} \right) \\
&\le \frac{2(i-1)}{2^{(d+\ell)n}} \cdot 2^{(d-1)n} + \sum_{x\in[1..\ell-1]} \left( \frac{2(i-1)}{2^{(d+x)n}} \cdot 2^{xn} \right) \\
&= \frac{2(i-1)}{2^{(\ell+1)n}} + \frac{2(\ell-1)(i-1)}{2^{dn}} .
\end{aligned}$$

This also holds true for $\ell = 1$, since we have for $q \le 2^n$ and $\ell = 1$,

$$\begin{aligned}
\Pr[i_s = i] &\le \sum_{j\in[1..i-1]} \left( \Pr[X_i^d = X_j^d] \cdot \Pr[X_i^{d+1} = X_j^{d+1}] \right) \\
&\le \frac{(i-1) - |\mathcal{X}_i^{d+1}|}{2^n - |\mathcal{X}_i^{d+1}|} \cdot \frac{2^{(d-1)n}}{2^{dn} - (i-1)} \\
&\le \frac{2(i-1)}{2^{(d+1)n}} \cdot 2^{(d-1)n} \\
&= \frac{2(i-1)}{2^{2n}} .
\end{aligned}$$

By following the same analysis, if the $i$-th query is an upper query, it holds for $q \le 2^n$ and $\ell \in [1..d-1]$ that

$$\Pr[i_s = i] \le \frac{2(i-1)}{2^{(\ell+1)n}} + \frac{2(\ell-1)(i-1)}{2^{dn}} .$$

Therefore, we have

$$\begin{aligned}
\Pr[T^{\mathrm{id}} \in \mathcal{T}^{\mathrm{bad}}] &\le \sum_{i\in[1..q_c+q_p]} \left( \frac{2(i-1)}{2^{(\ell+1)n}} + \frac{2(\ell-1)(i-1)}{2^{dn}} \right) \\
&\le \frac{(q_c+q_p)^2}{2^{(\ell+1)n}} + \frac{(\ell-1)(q_c+q_p)^2}{2^{dn}} .
\end{aligned}$$

□

**Evaluation of $\Pr[T^{\mathrm{re}} = \tau] / \Pr[T^{\mathrm{id}} = \tau]$.** We next consider $\tau \in \mathcal{T}^{\mathrm{good}}$. We prove the following lemma.

**Lemma 3.** *For any transcript $\tau \in \mathcal{T}^{\mathrm{good}}$, it holds that*

$$\frac{\Pr[T^{\mathrm{re}} = \tau]}{\Pr[T^{\mathrm{id}} = \tau]} \ge 1 - \frac{0.5(q_c+q_p)^2}{2^{dn}} .$$

*Proof.* We define

$$\mathcal{Q}^{\mathrm{upper}} = \{i \mid \mathcal{A}\text{'s } i\text{-th query to } \mathcal{O}_p \text{ is an upper query}\}$$

and

$$\mathcal{Q}^{\mathrm{lower}} = \{i \mid \mathcal{A}\text{'s } i\text{-th query to } \mathcal{O}_p \text{ is a lower query}\}\,.$$

Clearly, we have $|\mathcal{Q}^{\mathrm{upper}}| + |\mathcal{Q}^{\mathrm{lower}}| = q_c + q_p$.

In the ideal world, $\tau$ determines $q_c+q_p$ input-output pairs of $\pi$ and $(d+2\ell-1)\times(q_c+q_p)$ random variables each of which is chosen uniformly at random from the set of size $2^n - |\mathcal{X}_i^x|$ for $x \in [1..d+2\ell-1]$ and $i \in \mathcal{Q}^{\mathrm{upper}}$, and for $x \in [d+1..2d+2\ell-1]$ and $i \in \mathcal{Q}^{\mathrm{lower}}$. Therefore, we have

$$\begin{aligned}\Pr[T^{\mathrm{id}} = \tau] = &\left(\prod_{i\in[1..q_c+q_p]} \frac{1}{2^{dn}-(i-1)}\right)\cdot\left(\prod_{i\in\mathcal{Q}^{\mathrm{upper}}}\prod_{x\in[1..d+2\ell-1]} \frac{1}{2^n-|\mathcal{X}_i^x|}\right)\\ &\cdot\left(\prod_{i\in\mathcal{Q}^{\mathrm{lower}}}\prod_{x\in[d+1..2d+2\ell-1]} \frac{1}{2^n-|\mathcal{X}_i^x|}\right).\end{aligned}$$

In the real world, $\tau$ determines $q_c+q_p$ input-output pairs of $P_x$ for all $x \in [1..2d+2\ell-1]$. We define $\mathcal{Y}_i^x$ for $x \in [1..2d+2\ell-1]$ and $i \in [1..q_c+q_p]$ as

$$\mathcal{Y}_i^x = \{X_j^x \mid X_i^{[x+1..x+d-1]} = X_j^{[x+1..x+d-1]}\}\,.$$

Since $\tau \in \mathcal{T}^{\mathrm{good}}$ holds, we have $|\mathcal{X}_i^x| = |\mathcal{Y}_i^x|$ for $x \in [1..d+2\ell-1]$ and $i \in \mathcal{Q}^{\mathrm{upper}}$, and for $x \in [d+1..2d+2\ell-1]$ and $i \in \mathcal{Q}^{\mathrm{lower}}$. Therefore, we have

$$\begin{aligned}\Pr[T^{\mathrm{re}} = \tau] &= \prod_{i\in[1..q_c+q_p]}\prod_{x\in[1..2d+2\ell-1]} \frac{1}{2^n-|\mathcal{Y}_i^x|}\\ &\geq \left(\prod_{i\in\mathcal{Q}^{\mathrm{upper}}}\left(\prod_{x\in[1..d+2\ell-1]} \frac{1}{2^n-|\mathcal{X}_i^x|}\right)\cdot\left(\prod_{x\in[d+2\ell..2d+2\ell-1]} \frac{1}{2^n}\right)\right)\\ &\quad\cdot\left(\prod_{i\in\mathcal{Q}^{\mathrm{lower}}}\left(\prod_{x\in[1..d]} \frac{1}{2^n}\right)\cdot\left(\prod_{x\in[d+1..2d+2\ell-1]} \frac{1}{2^n-|\mathcal{X}_i^x|}\right)\right)\\ &= \left(\prod_{i\in[1..q_c+q_p]} \frac{1}{2^{dn}}\right)\cdot\left(\prod_{i\in\mathcal{Q}^{\mathrm{upper}}}\prod_{x\in[1..d+2\ell-1]} \frac{1}{2^n-|\mathcal{X}_i^x|}\right)\\ &\quad\cdot\left(\prod_{i\in\mathcal{Q}^{\mathrm{lower}}}\prod_{x\in[d+1..2d+2\ell-1]} \frac{1}{2^n-|\mathcal{X}_i^x|}\right).\end{aligned}$$

From the above, we have

$$\frac{\Pr[T^{\mathrm{re}} = \tau]}{\Pr[T^{\mathrm{id}} = \tau]} \geq \prod_{i\in[1..q_c+q_p]} \frac{2^{dn}-(i-1)}{2^{dn}} \geq 1 - \frac{0.5(q_c+q_p)^2}{2^{dn}}\,. \qquad \square$$

From Lemma 2, Lemma 3, and Lemma 1, we conclude the proof of Theorem 1.

# 6 Conclusions

We showed that the $(2d+2\ell-1)$-round $dn$-bit cryptographic permutation based on $2d+2\ell-1$ independent $n$-bit ideal ciphers with $\kappa n$-bit keys, where $d = \kappa + 1$, has an indifferentiability security bound of $O(q^2/2^{(\ell+1)n})$ under the assumption that $q \leq 2^n$, where $1 \leq \ell \leq d-1$. This implies that a 5-round version of Coron et al.'s construction and $(2d+1)$-round version of Guo and Lin's construction have an indifferentiability security bound of $O(q^2/2^{2n})$, and this also shows that the security bound of Guo and Lin's construction exponentially improves by adding every two more rounds, up to $4d-3$ rounds. To the best of our knowledge, these results give the first cryptographic permutation that is built from $n$-bit ideal ciphers and has a full $n$-bit indifferentiability security bound.

There are open questions. First, we do not know the tightness of our security bounds, e.g., we do not know if the 5-round version of Coron et al.'s construction and $(2d+1)$-round version of Guo and Lin's construction are differentiable with $O(2^n)$ queries. The tightness of the results in [CDMS10, GL15] is also unknown, and these are all left as an open question. We also do not know the security of a 4-round version of Coron et al.'s construction nor a $2d$-round version of Guo and Lin's construction. It would also be interesting to see if the condition of $q \leq 2^n$ can be removed from Theorem 1. Finally, all the result of this paper and those in [CDMS10, GL15] rely on the fact that ideal ciphers are independent, and it would be interesting to see the security of a construction where we use one ideal cipher in all rounds.

# Acknowledgments

The authors thank the anonymous reviewers of FSE 2021 for valuable comments that helped improving this paper. We also thank Frederik Armknecht for guidance. This work was supported in part by JSPS KAKENHI Grant Number 20K11675.

# References

[ADMA15] Elena Andreeva, Joan Daemen, Bart Mennink, and Gilles Van Assche. Security of keyed sponge constructions using a modular proof approach. In Gregor Leander, editor, *Fast Software Encryption - 22nd International Workshop, FSE 2015, Istanbul, Turkey, March 8-11, 2015, Revised Selected Papers*, volume 9054 of *Lecture Notes in Computer Science*, pages 364–384. Springer, 2015.

[BDH+17] Guido Bertoni, Joan Daemen, Seth Hoffert, Michaël Peeters, Gilles Van Assche, and Ronny Van Keer. Farfalle: parallel permutation-based cryptography. *IACR Trans. Symmetric Cryptol.*, 2017(4):1–38, 2017.

[BDPA08] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. On the indifferentiability of the sponge construction. In Nigel P. Smart, editor, *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*, volume 4965 of *Lecture Notes in Computer Science*, pages 181–197. Springer, 2008.

[BDPA10] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Sponge-based pseudo-random number generators. In Stefan Mangard and François-Xavier Standaert, editors, *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings*, volume 6225 of *Lecture Notes in Computer Science*, pages 33–47. Springer, 2010.

[BDPA11] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Duplexing the sponge: Single-pass authenticated encryption and other applications. In Ali Miri and Serge Vaudenay, editors, *Selected Areas in Cryptography - 18th International Workshop, SAC 2011, Toronto, ON, Canada, August 11-12, 2011, Revised Selected Papers*, volume 7118 of *Lecture Notes in Computer Science*, pages 320–337. Springer, 2011.

[BJK+16] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The SKINNY family of block ciphers and its low-latency variant MANTIS. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, volume 9815 of *Lecture Notes in Computer Science*, pages 123–153. Springer, 2016.

[BK09] Alex Biryukov and Dmitry Khovratovich. Related-key cryptanalysis of the full AES-192 and AES-256. In Mitsuru Matsui, editor, *Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings*, volume 5912 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2009.

[BKL+17] Daniel J. Bernstein, Stefan Kölbl, Stefan Lucks, Pedro Maat Costa Massolino, Florian Mendel, Kashif Nawaz, Tobias Schneider, Peter Schwabe, François-Xavier Standaert, Yosuke Todo, and Benoît Viguier. Gimli : A cross-platform permutation. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, volume 10529 of *Lecture Notes in Computer Science*, pages 299–320. Springer, 2017.

[BKN09] Alex Biryukov, Dmitry Khovratovich, and Ivica Nikolic. Distinguisher and related-key attack on the full AES-256. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, volume 5677 of *Lecture Notes in Computer Science*, pages 231–249. Springer, 2009.

[BLLN19] Arghya Bhattacharjee, Eik List, Cuauhtemoc Mancillas López, and Mridul Nandi. The Oribatida family of lightweight authenticated encryption schemes, version v1.2. Submission to the NIST Lightweight Cryptography Competition, 2019.

[BLN18] Ritam Bhaumik, Eik List, and Mridul Nandi. ZCZ - achieving $n$-bit SPRP security with a minimal number of tweakable-block-cipher calls. In Thomas Peyrin and Steven D. Galbraith, editors, *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part I*, volume 11272 of *Lecture Notes in Computer Science*, pages 336–366. Springer, 2018.

[BR04] Mihir Bellare and Phillip Rogaway. Code-based game-playing proofs and the security of triple encryption. *IACR Cryptol. ePrint Arch.*, 2004:331, 2004.

[CDMS10] Jean-Sébastien Coron, Yevgeniy Dodis, Avradip Mandal, and Yannick Seurin. A domain extender for the ideal cipher. In Daniele Micciancio, editor, *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010, Zurich,*

*Switzerland, February 9-11, 2010. Proceedings*, volume 5978 of *Lecture Notes in Computer Science*, pages 273–289. Springer, 2010.

[CHK+16] Jean-Sébastien Coron, Thomas Holenstein, Robin Künzler, Jacques Patarin, Yannick Seurin, and Stefano Tessaro. How to build an ideal cipher: The indifferentiability of the Feistel construction. *J. Cryptology*, 29(1):61–114, 2016.

[CLMP17] Yu Long Chen, Atul Luykx, Bart Mennink, and Bart Preneel. Efficient length doubling from tweakable block ciphers. *IACR Trans. Symmetric Cryptol.*, 2017(3):253–270, 2017.

[CMN18] Yu Long Chen, Bart Mennink, and Mridul Nandi. Short variable length domain extenders with beyond birthday bound security. In Thomas Peyrin and Steven D. Galbraith, editors, *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part I*, volume 11272 of *Lecture Notes in Computer Science*, pages 244–274. Springer, 2018.

[CPS08] Jean-Sébastien Coron, Jacques Patarin, and Yannick Seurin. The random oracle model and the ideal cipher model are equivalent. In David A. Wagner, editor, *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings*, volume 5157 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2008.

[CS14] Shan Chen and John P. Steinberger. Tight security bounds for key-alternating ciphers. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, volume 8441 of *Lecture Notes in Computer Science*, pages 327–350. Springer, 2014.

[CS15] Benoit Cogliati and Yannick Seurin. On the provable security of the iterated Even-Mansour cipher against related-key and chosen-key attacks. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 584–613. Springer, 2015.

[DKT16] Dana Dachman-Soled, Jonathan Katz, and Aishwarya Thiruvengadam. 10-round Feistel is indifferentiable from an ideal cipher. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 649–678. Springer, 2016.

[DN18] Avijit Dutta and Mridul Nandi. Tweakable HCTR: A BBB secure tweakable enciphering scheme. In Debrup Chakraborty and Tetsu Iwata, editors, *Progress in Cryptology - INDOCRYPT 2018 - 19th International Conference on Cryptology in India, New Delhi, India, December 9-12, 2018, Proceedings*, volume 11356 of *Lecture Notes in Computer Science*, pages 47–69. Springer, 2018.

[DR02] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Information Security and Cryptography. Springer, 2002.

[DS16] Yuanxi Dai and John P. Steinberger. Indifferentiability of 8-round Feistel networks. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 95–120. Springer, 2016.

[DSST17] Yuanxi Dai, Yannick Seurin, John P. Steinberger, and Aishwarya Thiruvengadam. Indifferentiability of iterated Even-Mansour ciphers with non-idealized key-schedules: Five rounds are necessary and sufficient. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III*, volume 10403 of *Lecture Notes in Computer Science*, pages 524–555. Springer, 2017.

[GL15] Chun Guo and Dongdai Lin. Improved domain extender for the ideal cipher. *Cryptography and Communications*, 7(4):509–533, 2015.

[HKT11] Thomas Holenstein, Robin Künzler, and Stefano Tessaro. The equivalence of the random oracle model and the ideal cipher model, revisited. In Lance Fortnow and Salil P. Vadhan, editors, *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 89–98. ACM, 2011.

[Min15] Kazuhiko Minematsu. Building blockcipher from small-block tweakable blockcipher. *Des. Codes Cryptography*, 74(3):645–663, 2015.

[MPS12] Avradip Mandal, Jacques Patarin, and Yannick Seurin. On the public indifferentiability and correlation intractability of the 6-round Feistel construction. In Ronald Cramer, editor, *Theory of Cryptography - 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy, March 19-21, 2012. Proceedings*, volume 7194 of *Lecture Notes in Computer Science*, pages 285–302. Springer, 2012.

[MRH04] Ueli M. Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In Moni Naor, editor, *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004, Proceedings*, volume 2951 of *Lecture Notes in Computer Science*, pages 21–39. Springer, 2004.

[NI19] Ryota Nakamichi and Tetsu Iwata. Iterative block ciphers from tweakable block ciphers with long tweaks. *IACR Trans. Symmetric Cryptol.*, 2019(4):54–80, 2019.

[NIS14] NIST Computer Security Division. SHA-3 standard: Permutation-based hash and extendable-output functions. FIPS Publication 202, National Institute of Standards and Technology, U.S. Department of Commerce, May 2014.

[Pat08] Jacques Patarin. The "Coefficients H" technique. In Roberto Maria Avanzi, Liam Keliher, and Francesco Sica, editors, *Selected Areas in Cryptography, 15th International Workshop, SAC 2008, Sackville, New Brunswick, Canada, August 14-15, Revised Selected Papers*, volume 5381 of *Lecture Notes in Computer Science*, pages 328–345. Springer, 2008.

[ST13] Thomas Shrimpton and R. Seth Terashima. A modular framework for building variable-input-length tweakable ciphers. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part I*, volume 8269 of *Lecture Notes in Computer Science*, pages 405–423. Springer, 2013.

# A Proof of Equation (10)

We prove for $q_c + q_p \leq 2^n$ that

$$\frac{(i-1) - \left(\sum_{y\in[1..x]} |\mathcal{X}_i^{d+y}|\right)}{\prod_{y\in[1..x]} (2^n - |\mathcal{X}_i^{d+y}|)} \cdot \frac{1}{2^{dn} - (i-1)} \leq \frac{2(i-1)}{2^{(d+x)n}}.$$

Here, $d \geq 2$, $\ell \in [1..d-1]$, $x \in [1..\ell]$, and $0 \leq \sum_{y\in[1..x]} |\mathcal{X}_i^{d+y}| < i \leq q_c + q_p \leq 2^n$.

*Proof.* We subtract the left hand side from the right hand side:

$$\begin{aligned}
&\frac{2(i-1)}{2^{(d+x)n}} - \frac{(i-1) - \left(\sum_{y\in[1..x]} |\mathcal{X}_i^{d+y}|\right)}{\prod_{y\in[1..x]} (2^n - |\mathcal{X}_i^{d+y}|)} \cdot \frac{1}{2^{dn} - (i-1)} \\
&\geq \frac{2(i-1) \cdot \left(2^{xn} - \left(\sum_{y\in[1..x]} |\mathcal{X}_i^{d+y}|\right) \cdot 2^{(x-1)n}\right) \cdot (2^{dn} - (i-1))}{2^{(d+x)n} \cdot \left(\prod_{y\in[1..x]} (2^n - |\mathcal{X}_i^{d+y}|)\right) \cdot (2^{dn} - (i-1))} \\
&\quad - \frac{\left((i-1) - \left(\sum_{y\in[1..x]} |\mathcal{X}_i^{d+y}|\right)\right) \cdot 2^{(d+x)n}}{2^{(d+x)n} \cdot \left(\prod_{y\in[1..x]} (2^n - |\mathcal{X}_i^{d+y}|)\right) \cdot (2^{dn} - (i-1))}.
\end{aligned}$$

Clearly, the denominator is positive. Therefore, we prove (the numerator) $\geq 0$. Let $\sigma_i = \sum_{y\in[1..x]} |\mathcal{X}_i^{d+y}|$, then we have

$$\begin{aligned}
(\text{the numerator}) &= 2(i-1) \cdot (2^n - \sigma_i) \cdot 2^{(x-1)n} \cdot (2^{dn} - (i-1)) - ((i-1) - \sigma_i) \cdot 2^{(d+x)n} \\
&= 2(i-1) \cdot 2^{(x-1)n} \cdot \left(2^{(d+1)n} - \sigma_i \cdot 2^{dn} - (2^n - \sigma_i)(i-1)\right) \\
&\quad - (i-1) \cdot 2^{(d+x)n} + \sigma_i \cdot 2^{(d+x)n} \\
&= (i-1) \cdot 2^{(x-1)n} \cdot \left(2^{(d+1)n} - 2\sigma_i \cdot 2^{dn} - 2(2^n - \sigma_i)(i-1)\right) \\
&\quad + \sigma_i \cdot 2^{(d+x)n} \\
&= (i-1) \cdot 2^{(x-1)n} \cdot \left((2^n - \sigma_i) \cdot 2^{dn} - 2(2^n - \sigma_i)(i-1)\right)
\end{aligned}$$

$$
\begin{aligned}
&\quad - (i-1) \cdot \sigma_i \cdot 2^{(d+x-1)n} + \sigma_i \cdot 2^{(d+x)n} \\
&= (i-1) \cdot 2^{(x-1)n} \cdot (2^n - \sigma_i) \left(2^{dn} - 2(i-1)\right) \\
&\quad + \sigma_i \cdot (2^n - (i-1)) \cdot 2^{(d+x-1)n} \\
&\geq 0\,. \qquad \square
\end{aligned}
$$