Revisiting the Indifferentiability of the Sum of Permutations

Aldo Gunsing[1], Ritam Bhaumik[2], Ashwin Jha[3], Bart Mennink[1], and Yaobin Shen[4]

[1] Digital Security Group, Radboud University, Nijmegen, The Netherlands
aldo.gunsing@ru.nl, b.mennink@cs.ru.nl
[2] EPFL, Switzerland
ritam.bhaumik@epfl.ch
[3] CISPA Helmholtz Center for Information Security, Saarbrücken, Germany
ashwin.jha@cispa.de
[4] UCLouvain, ICTEAM/ELEN/Crypto Group
yaobin.shen@uclouvain.be

Abstract. The sum of two n-bit pseudorandom permutations is known to behave like a pseudorandom function with n bits of security. A recent line of research has investigated the security of two public n-bit permutations and its degree of indifferentiability. Mandal et al. (INDOCRYPT 2010) proved $2n/3$-bit security, Mennink and Preneel (ACNS 2015) pointed out a non-trivial flaw in their analysis and re-proved $(2n/3 - \log_2(n))$-bit security. Bhattacharya and Nandi (EUROCRYPT 2018) eventually improved the result to n-bit security. Recently, Gunsing at CRYPTO 2022 already observed that a proof technique used in this line of research only holds for sequential indifferentiability. We revisit the line of research in detail, and observe that the strongest bound of n-bit security has two other serious issues in the reasoning, the first one is actually the same non-trivial flaw that was present in the work of Mandal et al., while the second one discards biases in the randomness influenced by the distinguisher. More concretely, we introduce two attacks that show limited potential of different approaches. We (i) show that the latter issue that discards biases only holds up to $2^{3n/4}$ queries, and (ii) perform a differentiability attack against their simulator in $2^{5n/6}$ queries. On the upside, we revive the result of Mennink and Preneel and show $(2n/3 - \log_2(n))$-bit regular indifferentiability security of the sum of public permutations.

Keywords: indifferentiability · sum of permutations · attacks · resolutions

1 Introduction

The question of how to achieve a secure pseudorandom function (PRF) from a pseudorandom permutation (PRP) has played a central role in symmetric cryptography. After all, we have the availability of many cryptographic primitives such as AES [10] that behave – or are at least claimed to behave – like

a pseudorandom permutation, whereas for, e.g., stream encryption or message authentication, we would like to have a primitive that behaves like a pseudorandom function. Dedicated pseudorandom functions, in turn, are scarce [1, 6, 24]. Instead, over the last decades, the question of PRF design has mostly been dominated by approaches of building them generically from PRPs.

An n-bit PRP behaves like an n-bit PRF, but only as long as the number of evaluations is below $2^{n/2}$, a result known as the PRP-PRF switch [3, 5, 8, 14, 16, 17]. As this birthday bound could be restrictive in case of small block ciphers, various beyond birthday bound constructions have been analyzed. One such construction is the sum of PRPs:

$$F_{K_0,K_1}(x) = \mathcal{E}(K_0, x) \oplus \mathcal{E}(K_1, x)\,,$$

where $\mathcal{E}$ is a PRP with a block size of n bits. The construction was first introduced by Bellare et al. [4]. Lucks [19] proved around $2^{2n/3}$ security, Bellare and Impagliazzo [2] around $2^n/n$ security, and Patarin [26–28] proved optimal 2^n security, up to constant, albeit using the mirror theory. Dai et al. [11] proved around 2^n security using the chi-squared method.

These results were all in the case where the underlying primitive was a PRP, i.e., a building block $\mathcal{E}$ that, when instantiated with a secret key, behaves like a secret random permutation. A natural related question is to what degree the sum of two *public* permutations behaves like a *public* random function. In other words, suppose we are given two n-bit permutations Π_0, Π_1, to what degree

$$F^{\Pi_0,\Pi_1}(x) = \Pi_0(x) \oplus \Pi_1(x)$$

behaves like a random function. As this function is keyless, we cannot rely on conventional indistinguishability (as we considered the sum of PRPs in), but instead, we should consider this function in the indifferentiability framework of Maurer et al. [22], or more specifically the version of Coron et al. [9] tailored towards symmetric cryptographic primitives. In this framework, one compares the function F in conjunction with the primitives Π_0, Π_1 with a random function $\mathcal{R}$ in conjunction with a simulator ensemble $\mathcal{S}_0, \mathcal{S}_1$, and one says that F *behaves like* $\mathcal{R}$ if there exists a simulator ensemble such that these two worlds are hard to distinguish.

In the indifferentiability framework, Mandal et al. [20] proved that the sum of permutations behaves like a random function up to $2^{n/2}$ queries, and even up to $2^{2n/3}$ queries with a slightly more involved simulator. Mennink and Preneel identified a flaw in the reasoning of Mandal et al. [20] and re-proved $(2n/3 - \log_2(n))$ security. Bhattacharya and Nandi [7] proved optimal 2^n indifferentiability of F, using a simulator that is slightly more involved than that of Mandal et al. and Mennink and Preneel. Lee [18] proved $2^{(r-1)n/r}$ security for the sum of r permutations, but only for even integers $r \geqslant 4$. Our focus is on the sum of two permutations. A more detailed description of the earlier security analyses is given in Section 3.

1.1 Issues With Existing Security Analyses

This state of the art suggests that the case of the sum of permutations is closed: there is a proof of optimal 2^n security both in the case of secret permutations as in the case of public permutations. However, nothing is further from the truth.

First of all, Gunsing [15] recently discussed a faulty reasoning in a proof technique used in the indifferentiability of tree hashing. In a nutshell, this proof technique consists of replacing the distinguisher by a slightly stronger distinguisher for which the security analysis was easier. The author also observed that the same technique was used for the indifferentiability of the sum of permutations. As a matter of fact, all four indifferentiability results on the sum of permutations [7, 18, 20, 25] use this proof technique. Concretely, it turns out that this proof technique *only* works in the case of sequential indifferentiability (i.e., where the distinguisher should first make its primitive queries before making construction queries, cf., Section 2.3). Concluding, all indifferentiability results on the sum of permutations known to date *only hold for sequential indifferentiability*. This issue of 'sequentiality' is discussed in detail in Section 3.1.

Inspired by this, we aimed to fix the proof of Bhattacharya and Nandi [7] to reattain 2^n (regular) indifferentiability, but while doing so, we observed that this was not easily done. To the contrary, we observed two additional issues in the security proof of Bhattacharya and Nandi that made it impossible for us to reattain optimal 2^n security.

The first issue, dubbed 'fresh oracle', is about the fact that the proof assumes that any query to the random oracle returns a uniform random value. This is not the case when the inverse simulator tries a value and rejects it when it is not a suitable value, as this rejection leads to a bias in future values. The ideal world could be modified to have the more uniform behavior, but this comes at the cost of $3n/4$-bit security, diminishing the optimal n-bit security bound. We show this as an attack described in Section 5. Interestingly, this difference does not lead to an attack on the real construction, as the real world behaves more like the regular ideal world and not the modified one. So this bias is a feature, not a bug, and it is necessary to prove more than $3n/4$-bit security, even in the sequential setting.

The second issue, dubbed 'random range', centers around the problem that the proof assumes that the ranges of the primitives are randomly sampled. This is not the case as the adversary can freely choose this by making inverse queries with a desired range. This is a fundamental error that basically ignores the inverse queries. (The problem is of the same vein as what Mennink and Preneel identified for the proof of Mandal et al., though more ingenious.) The issue is not solvable, except by removing the inverse query direction. The issue is discussed in detail in Section 3.3.

To conclude, whereas Gunsing [15] already suggested that the entire line of research on the indifferentiability of the sum of permutations only holds in the case of sequential indifferentiability, we go even further to state that the proof of 2^n indifferentiability contains two fundamental and seemingly unsolvable gaps. A summary of these issues is given in Table 1. In short, the only result remaining

that is not fundamentally problematic is that of Mennink and Preneel [25], which proves $(2n/3 - \log_2(n))$ bits of security, but its result has to be reduced to the weaker sequential indifferentiability setting. There is no known result for the normal indifferentiability setting.

Table 1. Overview of the previous results with the claimed security level and the different errors they contain.

paper	security level	sequentiality (Section 3.1)	fresh oracle (Section 3.2)	random range (Section 3.3)
[20]	$2n/3$	[15]	—	[25]
[25]	$2n/3 - \log_2(n)$	[15]	—	—
[7]	n	[15]	now	now
Section 4	$2n/3 - \log_2(n)$	—	—	—

1.2 New Indifferentiability Proof

Taking all issues in earlier analyses into account, the best result to date is $2^{2n/3}/n$ sequential indifferentiability, i.e., the result of Mennink and Preneel [25] but only for the case of sequential indifferentiability. As this state of affairs is rather unsatisfactory, we reconsider the (regular) indifferentiability of the sum of permutations in Section 4, and manage to prove $2^{2n/3}/n$ (regular) indifferentiability. The idea of the proof is to extend the analysis of Mennink and Preneel. We make use of its result in the sequential setting as a black box and extend it to the (regular) indifferentiability setting by proving that queries of the simulator can be swapped at a cost of $2^{2n/3}$, maintaining the same bound. The new security result is also included in Table 1.

1.3 Generic Attack

Inspired by the fact that we managed to restore the proof of Mennink and Preneel [25] in the (regular) indifferentiability setting, one might be tempted to investigate the possibility to restore the proof of Bhattacharya and Nandi [7] in the (regular) indifferentiability setting. However, it turns out that this is not possible: in Section 5 we describe an attack in $2^{3n/4}$ queries that show that the 'fresh oracle' simplification cannot be made and in Section 6 we describe a generic differentiability attack in $2^{5n/6}$ queries, that succeeds against the simulator of Bhattacharya and Nandi, which is in fact the most logical choice of simulator and all previous works use a variant of it.

Our generic attack implies that the best 'one can hope for' is $2^{5n/6}$ indifferentiability, except for possibly expanding the simulator to a much more advanced one. Admittedly, this still exposes a gap between the $2^{2n/3}/n$ indifferentiability bound of Section 4, but it turns out (as also testified by the multiple issues found in earlier analyses) that proving tightness for the indifferentiability of the sum

of permutations is very hard, and we leave tightness of the indifferentiability of the sum of permutations as an open problem.

1.4 Applications

Besides being of general theoretical interest, the sum of two public random permutations has important implications for the design of cryptographic schemes. Examples include as a building block to construct beyond-birthday-bound domain extender [23], as a building block to construct collision-resistant compression function [29,30], and as a building block to construct variable input length random oracle [12].

2 Preliminaries

2.1 Notation

Let $n \geqslant 1$ be an integer. Let $\{0,1\}^n$ be the set of all n-bit strings. Let $\mathrm{func}(n)$ be the set of all functions from $\{0,1\}^n \to \{0,1\}^n$ and $\mathrm{perm}(n)$ the set of permutations on $\{0,1\}^n$. For a set $\mathcal{X}$, we denote by $x \xleftarrow{\$} \mathcal{X}$ the uniformly random sampling of an element from $\mathcal{X}$. If x and y are two bit-strings of the same length, we denote by $x \oplus y$ their bit-wise XOR.

2.2 Sum of Permutations

We will restrict our focus to the sum of two independent public permutations.[5] Let $\Pi_0, \Pi_1 \in \mathrm{perm}(n)$ be two n-bit permutations. The sum of permutations is the construction $F : \{0,1\}^n \to \{0,1\}^n$ defined as

$$F^{\Pi_0,\Pi_1}(x) = \Pi_0(x) \oplus \Pi_1(x)\,. \tag{1}$$

The output of Π_0 will typically be denoted y_0, the output of Π_1 will typically be denoted y_1, and the output of F will be denoted $z = y_0 \oplus y_1$. See also Figure 1. In the remainder, we will drop the superscript access of F for brevity.

2.3 Indifferentiability

Maurer et al. [22] introduced indifferentiability as an extension of indistinguishability, in order to measure the degree in which a keyless function behaves like its random counterpart. Coron et al. [9] applied the model to cryptographic hash functions, and we will adopt their model. In fact, in our work, we will restrict our focus to the construction $F : \{0,1\}^n \to \{0,1\}^n$ built on top of two permutations $\Pi_0, \Pi_1 : \{0,1\}^n \to \{0,1\}^n$.

Before stating what we mean when a construction is indifferentiable, we first pose the differentiability setup.

[5] It is possible to describe a variant based on one permutation $\Pi \in \mathrm{perm}(n)$ using domain separation, as in $\Pi(x\|0) \oplus \Pi(x\|1)$.

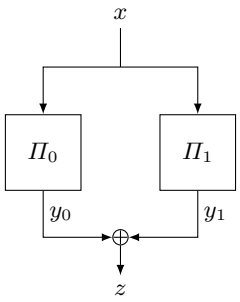

Fig. 1. The sum of permutations.

Definition 1 (Differentiability Setup). *Let* $F : \{0,1\}^n \to \{0,1\}^n$ *be the construction of* (1) *based on ideal permutations* $\Pi_0, \Pi_1 \xleftarrow{\$} \mathrm{perm}(n)$*. Denote* $\Pi = (\Pi_0, \Pi_1)$ *for brevity. Let* $\mathcal{R} \xleftarrow{\$} \mathrm{func}(n)$ *be a random function with the same domain and range as* F*. Let* $\mathcal{S} = (\mathcal{S}_0, \mathcal{S}_1)$ *be a simulator with the same domain and range as* $\Pi = (\Pi_0, \Pi_1)$ *that has access to* $\mathcal{R}$*. The advantage of an indifferentiability distinguisher* $\mathcal{D}$ *against* F *with respect to simulator* $\mathcal{S}$ *is defined as*

$$\mathbf{Adv}_{F,\mathcal{S}}(\mathcal{D}) = \left| \mathbb{P}\left[\mathcal{D}^{F,\Pi,\Pi^{-1}} = 1\right] - \mathbb{P}\left[\mathcal{D}^{\mathcal{R},\mathcal{S},\mathcal{S}^{-1}} = 1\right] \right| . \tag{2}$$

The differentiability setup is depicted in Figure 2. We will refer to (F, Π, Π^{-1}) as the *real world* and to $(\mathcal{R}, \mathcal{S}, \mathcal{S}^{-1})$ as the *ideal world*. The attacker can make a *construction* query to $\mathcal{C}$ (F in the real world and $\mathcal{R}$ in the ideal world) and it can make a *primitive* query to $\mathcal{P}$ ($\Pi^{\pm}$ in the real world and $\mathcal{S}^{\pm}$ in the ideal world).

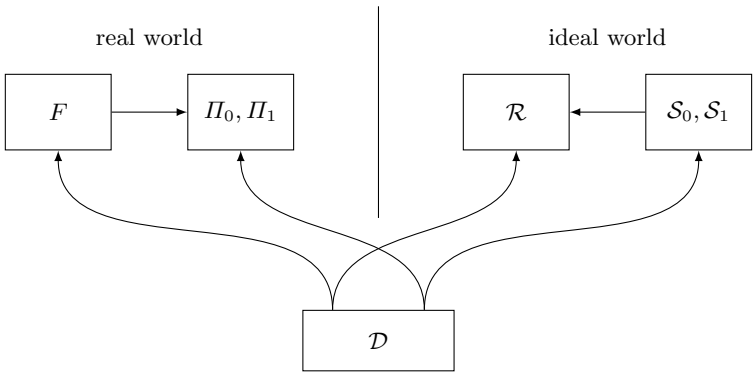

Fig. 2. The indifferentiability setup.

In (regular) indifferentiability as formalized by Coron et al. [9], the distinguisher has full freedom in the order in which it makes the queries.

Definition 2 ((Regular) Indifferentiability). *The construction* $F : \{0,1\}^n \to \{0,1\}^n$ *of* (1) *built on ideal permutations* $\Pi_0, \Pi_1 \xleftarrow{\$} \mathrm{perm}(n)$ *is* regularly ε-*indifferentiable from a random oracle* $\mathcal{R} \xleftarrow{\$} \mathrm{func}(n)$ *if there exists a simulator* $\mathcal{S} = (\mathcal{S}_0, \mathcal{S}_1)$ *such that*

$$\mathbf{Adv}^{\mathrm{indif}}_{F,\mathcal{S}}(\mathcal{D}) < \varepsilon$$

for any distinguisher $\mathcal{D}$ *that can make its construction and primitive queries in a fully adaptive manner.*

We will also discuss a weaker variant of indifferentiability, called public indifferentiability as introduced by Yoneyama et al. [31] and Dodis et al. [13]. Here, the queries made by the distinguisher to the construction oracle are *public* and known to the simulator. Canonically, the simulator will internally execute its own queries corresponding to the given input to deduce the result given by the construction oracle.

Definition 3 (Public Indifferentiability). *A construction* $F : \{0,1\}^n \to \{0,1\}^n$ *of* (1) *built on ideal permutations* $\Pi_0, \Pi_1 \xleftarrow{\$} \mathrm{perm}(n)$ *is* publicly ε-*indifferentiable from a random oracle* $\mathcal{R} \xleftarrow{\$} \mathrm{func}(n)$ *if there exists a simulator* $\mathcal{S} = (\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_{\mathrm{con}})$ *such that*

$$\mathbf{Adv}^{\text{seq-indif}}_{F,\mathcal{S}}(\mathcal{D}) < \varepsilon$$

for any distinguisher $\mathcal{D}$ *that can make its construction and primitive queries in a fully adaptive manner. The procedure* $\mathcal{S}_{\mathrm{con}}(x)$ *is executed whenever* $\mathcal{D}$ *makes the query* $\mathcal{R}(x)$*.*

We also look at another variant, sequential indifferentiability as introduced by Mandal et al. [21]. Sequential indifferentiability differs from (regular) indifferentiability only in the sense that the distinguisher *cannot make its queries in a fully adaptive manner*. Instead, it is restricted to *first* making its primitive queries and *then* its construction queries. It turns out that sequential indifferentiability is equivalent to public indifferentiability for stateless ideal primitives [21], which includes the sum of permutations.

Definition 4 (Sequential Indifferentiability). *A construction* $F : \{0,1\}^n \to \{0,1\}^n$ *of* (1) *built on ideal permutations* $\Pi_0, \Pi_1 \xleftarrow{\$} \mathrm{perm}(n)$ *is* sequentially ε-*indifferentiable from a random oracle* $\mathcal{R} \xleftarrow{\$} \mathrm{func}(n)$ *if there exists a simulator* $\mathcal{S} = (\mathcal{S}_0, \mathcal{S}_1)$ *such that*

$$\mathbf{Adv}^{\text{seq-indif}}_{F,\mathcal{S}}(\mathcal{D}) < \varepsilon$$

for any distinguisher $\mathcal{D}$ *that is restricted to first making its primitive queries and then making its construction queries.*

Algorithm 1 Definition of the uniform simulator with parameter ℓ

function $\mathcal{S}_b(x)$
  if $x \in \mathsf{domain}(\mathsf{P}_b)$ **then**
    return $\mathsf{P}_b(x)$
  $z \leftarrow \mathcal{R}(x)$
  $y_b \xleftarrow{\$} \{0,1\}^n \setminus (\mathsf{range}(\mathsf{P}_b) \cup (\mathsf{range}(\mathsf{P}_{1-b}) \oplus z))$
  $\mathsf{P}_b(x) \leftarrow y_b$
  $\mathsf{P}_{1-b}(x) \leftarrow y_b \oplus z$
  return y_b

function $\mathcal{S}_b^{-1}(y_b)$
  if $y_b \in \mathsf{range}(\mathsf{P}_b)$ **then**
    return $\mathsf{P}_b^{-1}(y_b)$
  for ℓ times **do**
    $x \xleftarrow{\$} \{0,1\}^n \setminus \mathsf{domain}(\mathsf{P}_0)$
    $z \leftarrow \mathcal{R}(x)$
    if $y_b \oplus z \notin \mathsf{range}(\mathsf{P}_{1-b})$ **then**
      $\mathsf{P}_b(x) \leftarrow y_b$
      $\mathsf{P}_{1-b}(x) \leftarrow y_b \oplus z$
      return x
  return $\perp$

Clearly, sequential indifferentiability is a weaker variant than (regular) indifferentiability in the sense that it significantly restricts the power of the distinguisher. Intuitively, in sequential indifferentiability, the queries that matter most are the primitive queries, and the construction queries made afterwards are only made to verify consistency in the primitive queries. The distinguisher has no possibility to use these construction queries to smartly select upcoming primitive queries.

The definition of indifferentiability requires the existence of a simulator $\mathcal{S}$. For a lower bound this means that providing an explicit one is sufficient. However, for an upper bound one would have to show attacks for *any* simulator, which is very difficult to do. Instead, for our attacks we focus on the most logical definition of the simulator, which we call the uniform simulator as it selects its values uniformly at random. It has a parameter ℓ determining how many times a loop should be executed. All previous works [7, 20, 25] use essentially this simulator with varying ℓ.

Definition 5 (Uniform Simulator). *The* uniform simulator *with parameter ℓ is defined in Algorithm 1.*

The uniform simulator internally keeps two partial permutations P_0 and P_1 consisting of the previously made queries. Additionally, on input $\mathcal{S}_b(x)$ (with $b \in \{0,1\}$) it not only sets $\mathsf{P}_b(x)$, but also $\mathsf{P}_{1-b}(x)$ and similarly for $\mathcal{S}_b^{-1}(y_b)$ it sets both $\mathsf{P}_b(x) = y_b$ and $\mathsf{P}_{1-b}(x)$ for some x.

The forward simulator simply samples uniformly from all possible outputs $\{0,1\}^n \setminus (\mathsf{range}(\mathsf{P}_b) \cup (\mathsf{range}(\mathsf{P}_{1-b}) \oplus z))$, where b denotes the selected permutation and $z = \mathcal{R}(x)$ is the output of random oracle for the input x.

The backward simulator is slightly more involved. It chooses a new input x uniformly at random (note that $\mathsf{domain}(\mathsf{P}_0) = \mathsf{domain}(\mathsf{P}_1)$), consults the random oracle to get $z = \mathcal{R}(x)$ and checks whether this x is possible as the condition $y_b \oplus z \notin \mathsf{range}(\mathsf{P}_{1-b})$ is required. It repeats this process up to ℓ times. If there is still no suitable x found, the simulator fails by returning $\perp$.

The parameter ℓ determines how many times the simulator tries an x. If $\ell = 1$ the simulator never retries, leading to a failure probability equal to the birthday bound. For larger ℓ the simulator can try multiple times, making the failure probability smaller and smaller. On the other hand, ℓ should not be too large as the simulator makes at most ℓq queries to the random oracle for q primitive queries. The first two works [20, 25] set $\ell = 2$ and [7] sets $\ell = n$.

Remark 1. The simulator given in [7] is slightly different, as they do not resample previously selected guesses: let x_i be chosen at iteration i of the loop ($1 \leqslant i \leqslant \ell$), then $x_i \xleftarrow{\$} \{0,1\}^n \setminus (\mathsf{domain}(\mathsf{P}_0) \cup \{x_1, \ldots, x_{i-1}\})$. This difference is negligible and only influences the failure probability. In fact, the selection of the parameter ℓ does not influence the outputs of the simulator at all, conditioned on the fact that it did not fail. Only the failure probability is impacted, which has a probability of $\mathcal{O}\left(q^{\ell+1}/2^{\ell n}\right)$.

3 Earlier Security Analysis

The sum of *secret* permutations has a long history, dating back to Impagliazzo and Rudich in 1988 [17]. A long sequence of research [2, 4, 11, 19, 26–28] has lead to a final conclusion that the sum of two *secret* permutations is hard to distinguish from a random function up to 2^n queries.

In this work, we are however concerned with the sum of *public* permutations, a problem that is more recent. In 2010, Mandal et al. [20] gave two indifferentiability results, one proving $2^{n/2}$ with a naive simulator (the simulator of Definition 5 with $\ell = 1$), and one result with a more involved simulator (the simulator of Definition 5 with $\ell = 2$) achieving $2^{2n/3}$ security. However, later, Mennink and Preneel observed that the latter result was flawed. In detail, the analysis of Mandal et al. relied on the premise that if the distinguisher makes q primitive queries, for any value $z \in \{0,1\}^n$ there are not more than $O(q^2/2^n)$ tuples $\{(x_0, y_0), (x_1, y_1)\}$ satisfying $y_0 \oplus y_1 = z$, a premise that was obviously false as the distinguisher can make inverse queries to the primitive. (Refer to [25, Section 4.3] for a more detailed discussion of the issue.) A noteworthy proof technique used by Mandal et al. was that the proof started with a transition of the distinguisher $\mathcal{D}$ to a more powerful distinguisher $\mathcal{D}'$ to make the security analysis easier (this transition is explained in more detail in Section 3.1).

In 2015, Mennink and Preneel re-proved $2^{2n/3}/n$ security [25], with the same simulator (the simulator of Definition 5 with $\ell = 2$) and a comparable proof technique as Mandal et al., but with a different bad event in the security analysis. In 2018, Bhattacharya and Nandi [7] proved optimal 2^n indifferentiability of F. The simulator of Bhattacharya and Nandi only marginally changed from the

simulator of Mandal et al., the only difference being that the simulator would potentially redraw up to $\ell = n$. In 2017, Lee [18] proved $2^{(r-1)n/r}$ security for the sum of r permutations for even integers $r \geqslant 4$.

It is important to mention that all these results adopted the proof technique used by Mandal et al. to start the proof by replacing the distinguisher $\mathcal{D}$ by a more powerful distinguisher $\mathcal{D}'$, or a similar technique, to make the security analysis easier. In the context of tree hashing, Gunsing [15] pointed out that this reasoning is faulty, and only holds in the case of sequential indifferentiability. We elaborate on this 'sequentiality' in detail in Section 3.1. In addition, we observe two additional issues in the proof of Bhattacharya and Nandi, namely the 'fresh oracle' problem (in Section 3.2) and the 'random range' problem (in Section 3.3).

3.1 Sequentiality

As shown in [15] all known results only hold for sequential indifferentiability. We explain the problem specifically for the case of the sum of permutations. In all previous works [7, 18, 20, 25] the error appears in a similar form.

Moving from $\mathcal{D}$ to $\mathcal{D}'$. The works [20, 25] make an explicit modification to the distinguisher. For any distinguisher $\mathcal{D}$ another distinguisher $\mathcal{D}'$ is constructed that behaves as follows:

1. Interact like $\mathcal{D}$;
2. At the end of the interaction, query $\mathcal{P}_0(x)$ and $\mathcal{P}_1(x)$ for any construction query $\mathcal{C}(x)$ made in the previous step (if not already done);
3. Output the same decision as $\mathcal{D}$.

As $\mathcal{D}'$ outputs the same decision as $\mathcal{D}$ and its extra queries happen at the end of the interaction, its advantage is the same as that of $\mathcal{D}$. Furthermore, as $\mathcal{C}(x)$ can be derived from $\mathcal{P}_0(x)$ and $\mathcal{P}_1(x)$, we can ignore the construction queries from the transcript and focus just on the primitive queries of the form $\mathcal{P}_0(x)$ and $\mathcal{P}_1(x)$. However, this last reasoning where the construction queries are ignored is incorrect and cannot be done. It ignores the fact that future queries can *depend* on the output of these construction queries. For example, let $\mathcal{D}$ be the distinguisher that generates two arbitrary inputs x_1 and x_2 and interacts as follows:

1. Make the construction query $z_1 = \mathcal{C}(x_1)$;
2. Make the construction query $z_2 = \mathcal{C}(x_2)$;
3. Compare z_1 and z_2 lexicographically and define $x^{\min}$ and $x^{\max}$ as:
   - If $z_1 \leqslant z_2$, then $x^{\min} = x_1$ and $x^{\max} = x_2$;
   - Otherwise, $x^{\min} = x_2$ and $x^{\max} = x_1$;
4. Make the primitive queries $y_0^{\min} = \mathcal{P}_0(x^{\min})$ and $y_1^{\min} = \mathcal{P}_1(x^{\min})$;
5. Make the primitive queries $y_0^{\max} = \mathcal{P}_0(x^{\max})$ and $y_1^{\max} = \mathcal{P}_1(x^{\max})$.

Here, the final transcript looks like

$$\Big((x_1, \mathcal{C}(x_1)), (x_2, \mathcal{C}(x_2)), (x^{\min}, \mathcal{P}_0(x^{\min}), \mathcal{P}_1(x^{\min})), (x^{\max}, \mathcal{P}_0(x^{\max}), \mathcal{P}_1(x^{\max}))\Big).$$

While it is the case that $\mathcal{C}(x_1) = \mathcal{P}_0(x_1) \oplus \mathcal{P}_1(x_1)$ and $\mathcal{C}(x_2) = \mathcal{P}_0(x_2) \oplus \mathcal{P}_1(x_2)$ can be derived from the full transcript (as x_1 and x_2 are either $x^{\min}$ or $x^{\max}$), they cannot simply be dropped, simplifying the transcript to

$$\Big((x^{\min}, \mathcal{P}_0(x^{\min}), \mathcal{P}_1(x^{\min})), (x^{\max}, \mathcal{P}_0(x^{\max}), \mathcal{P}_1(x^{\max}))\Big),$$

as this transcript is not well-defined. The input to the first query is $x^{\min}$, but this definition only makes sense given $\mathcal{C}(x_1)$ and $\mathcal{C}(x_2)$, whose values are still unknown.

One way to salvage the results is to consider the weaker notion of sequential indifferentiability, where all primitive queries have to be made before the construction queries. In this setting this dependence is not present as the construction queries happen last. Hence, we can downgrade a proof containing this flaw to the sequential indifferentiability setting.

Public Construction Queries. The works [7, 18] do not make an explicit modification to $\mathcal{D}$ but they make a similar mistake. Again, at the end of the interaction we give the outputs $\mathcal{P}_0(x)$ and $\mathcal{P}_1(x)$ for every made construction query $\mathcal{C}(x)$. Furthermore, if $\mathcal{P}_0(x)$ is made, then $\mathcal{P}_1(x)$ is given and vice versa. This means that for query i, the tuple

$$(x_i, \mathcal{P}_0(x_i), \mathcal{P}_1(x_i))$$

is known, from which the construction output $\mathcal{C}(x_i) = \mathcal{P}_0(x_i) \oplus \mathcal{P}_1(x_i)$ can be derived immediately. However, this step implicitly reorders some primitive queries. Consider the same interaction as before, where the final transcript looks like

$$\Big((x_1, \mathcal{C}(x_1)), (x_2, \mathcal{C}(x_2)), (x^{\min}, \mathcal{P}_0(x^{\min}), \mathcal{P}_1(x^{\min})), (x^{\max}, \mathcal{P}_0(x^{\max}), \mathcal{P}_1(x^{\max}))\Big).$$

Now, with the 'additional' information added this becomes

$$\Big((x_1, \mathcal{P}_0(x_1), \mathcal{P}_1(x_1)), (x_2, \mathcal{P}_0(x_2), \mathcal{P}_1(x_2))\Big).$$

Whenever $x^{\min} = x_2$, this reorders the primitive queries made. In the real world this does not matter, but in the ideal world the primitive is a simulator for which the order can be important. In general, for stateless primitives, like random functions and permutations, the order of the queries does not matter as there is no state to be influenced. However, most simulators are stateful, in which case the queries influence the state and with that the distribution of future queries as well. Ideally, the simulator should behave like a stateless primitive as much as possible. We quantify this notion by looking at what influence swapping two queries has

on the distribution of the outputs. The core of our new indifferentiability proof in Section 4 is to show that we can reorder the simulator queries up to $\mathcal{O}\left(2^{2n/3}\right)$ queries. Additionally, we also show an impossibility result in Section 6 where we make use of the fact that for the uniform simulator the order does matter, leading to an attack on it using $\mathcal{O}\left(2^{5n/6}\right)$ queries.

This reordering of the simulator queries has more in common with the notion of public indifferentiability. In this setting, which is equivalent to the previously mentioned sequential indifferentiability, the queries made by the distinguisher to the construction are publicly available to the simulator. The simplest way the simulator can make use of this is by making the queries $\mathcal{S}_0(x)$ and $\mathcal{S}_1(x)$ internally whenever the distinguisher queries $\mathcal{R}(x)$. This does correspond to the transformation made in [7], but it only holds for public indifferentiability. Note that it is true that the simulator can internally execute $\mathcal{S}_1(x)$ whenever $\mathcal{S}_0(x)$ is made as this is a query to the simulator. However, it can only execute these when $\mathcal{R}(x)$ happens if the construction queries are public. This is the case for public indifferentiability, but in (regular) indifferentiability they are not.

Reording Simulator Queries. We use this reordering idea in our new indifferentiability proof. In essence, we use many intermediate worlds in which the verification are step-wise moved to the end to convert the transcript corresponding to the public indifferentiability setting to the regular one. Instead of directly verifying the construction queries, we delay by a predetermined number of queries. We illustrate this using the example above. We start with the world corresponding to the case of public indifferentiability, where the verification queries of construction queries happen immediately. We denote them just before the construction queries, but this does not matter and is an arbitrary choice.

$$\Big((x_1, \mathcal{P}_0(x_1), \mathcal{P}_1(x_1)), (x_1, \mathcal{C}(x_1)), (x_2, \mathcal{P}_0(x_2), \mathcal{P}_1(x_2)), (x_2, \mathcal{C}(x_2)),$$
$$(x^{\min}, \mathcal{P}_0(x^{\min}), \mathcal{P}_1(x^{\min})), (x^{\max}, \mathcal{P}_0(x^{\max}), \mathcal{P}_1(x^{\max}))\Big),$$

where we do denote duplicate queries. We start with delaying the second verification query. We do this one step at a time. After three steps (one swap with the construction query, two with primitive queries) it will be verified at the end:

$$\Big((x_1, \mathcal{P}_0(x_1), \mathcal{P}_1(x_1)), (x_1, \mathcal{C}(x_1)), (x_2, \mathcal{C}(x_2)), (x^{\min}, \mathcal{P}_0(x^{\min}), \mathcal{P}_1(x^{\min})),$$
$$(x^{\max}, \mathcal{P}_0(x^{\max}), \mathcal{P}_1(x^{\max})), (x_2, \mathcal{P}_0(x_2), \mathcal{P}_1(x_2))\Big).$$

Now that the second query is completed, we move to the first query. After four delaying steps (two over a construction queries two over primitive queries), we get the following:

$$\Big((x_1, \mathcal{C}(x_1)), (x_2, \mathcal{C}(x_2)), (x^{\min}, \mathcal{P}_0(x^{\min}), \mathcal{P}_1(x^{\min})),$$

$$(x^{\max}, \mathcal{P}_0(x^{\max}), \mathcal{P}_1(x^{\max})), (x_1, \mathcal{P}_0(x_1), \mathcal{P}_1(x_1)), (x_2, \mathcal{P}_0(x_2), \mathcal{P}_1(x_2))\Big).$$

This corresponds to the normal transcript resulting from the indifferentiability setting, completing the process. We use the same idea in our proof, in reverse, to convert a normal transcript to one used in public indifferentiability. Furthermore, as there the construction queries can be derived from the primitive queries, they can essentially be ignored. In general they still have to be executed at the end to verify consistency, resulting in a sequential transcript. What is left to show is that this swapping of simulator queries has limited influence on its output distribution, which is a major part of the proof requiring a lot of computation.

3.2 Fresh Oracle

Even if we still consider [7] and restrict our focus to sequential indifferentiability, there is another problem, giving at most $3n/4$ bits of security. The problem is that for every query in the ideal world, the output of the construction oracle $\mathcal{R}$ is considered to be uniformly random. However, the inverse simulator can have a candidate output x, query the random oracle $z = \mathcal{R}(x)$, reject x as an output based on z and continue with a new candidate x'. This means that when $\mathcal{R}(x)$ for the same candidate x is queried later, its output will not be uniformly random, as it was rejected earlier and it is known what values are rejected. While it is not known what the candidate outputs were, it will still give a bias. Consider the following example:

1. Suppose there is some earlier interaction, giving non-empty domain D and ranges R_0 and R_1, all of size q. There is no bias in $\mathcal{R}$ yet;
2. Make the inverse query $x_1 = \mathcal{S}_0^{-1}(y_1)$ for an arbitrary $y_1 \notin R_0$;
3. Make the construction query $z_2 = \mathcal{R}(x_2)$ for an arbitrary $x_2 \notin D \cup \{x_1\}$.

In [7] the distribution of z_2 is considered to be uniformly random over $\{0,1\}^n$. However, there is the possibility that x_2 was a rejected candidate in step (2), making the probability that $z_2 \in R_1 \oplus y_1$ slightly more likely. More precisely, we can consider the outputs of $\mathcal{R}(x)$ to be determined just before step (2). As all values are independently and uniformly chosen, the probability that $\mathcal{R}(x) \in R_1 \oplus y_1$ is $q/2^n$. For $X = \{\, x \in \{0,1\}^n \setminus D \mid \mathcal{R}(x) \in R_1 \oplus y_1 \,\}$ we have $\mathbb{E}[|X|] = (2^n - |D|)q/2^n$. Before step (2) there are trivially $|X|$ values $x \in \{0,1\}^n \setminus D$ such that $\mathcal{R}(x) \in R_1 \oplus y_1$. After step (2) this value stays the same, because $x_1 \notin X$ as the simulator would reject x_1 otherwise and redraw. Therefore there are still $|X|$ values in $\{0,1\}^n \setminus D \cup \{x_1\}$ such that $\mathcal{R}(x) \in R_1 \oplus y_1$ while the size of possible values for x decreased by one. This means that at step (3), conditioned on that the simulator did not fail, we have that

$$\begin{aligned}\mathbb{P}[z_2 \in R_1 \oplus y_1] &= \mathbb{E}\left[\frac{|X|}{2^n - |D| - 1}\right] = \frac{(2^n - |D|)q}{2^n(2^n - |D| - 1)} \\ &= \frac{q}{2^n}\frac{2^n - q}{2^n - q - 1} = \frac{q}{2^n}\left(1 + \frac{1}{2^n - q - 1}\right),\end{aligned}$$

which is slightly higher than the uniform probability of $q/2^n$. Note that it actually does not matter for the distribution how many times the simulator retries. Given the fact that it did not fail, the probabilities will always be the same. The only influence that the number of retries gives is the probability for failure, which is roughly $q^{\ell+1}/2^{\ell n}$ for ℓ attempts.

As a consequence, z_2 is not uniformly distributed but has a slight positive bias towards the set $R_1 \oplus y$ and a slight negative bias towards the other values. The earlier works [20, 25] circumvent this problem by adding a bad event for repeated evaluations of $\mathcal{R}(x)$, contributing to limiting the proven security level to $(2n/3 - \log_2(n))$ bits. A way to view the flaw is that the proof in [7] does not consider the regular ideal world, but that it implicitly considers a modified ideal world. This modified world, which we will call 'fresh ideal world', is similar to the regular ideal world, but when the random function $\mathcal{R}(x)$ is asked for an output on input x, it will always output a uniformly random value from $\{0,1\}^n$, even when it was queried x before. The only exception is when it was queried by the distinguisher or when it can be derived from the primitive queries, then it will be consistent with the previous values. We formalize this in Algorithm 2, where the modified random function is denoted by $\mathcal{R}'$ and the modified simulator by $\mathcal{S}'$. The latter is based on the existing simulator $\mathcal{S}[\mathcal{R}']$, with its oracle access to $\mathcal{R}'$ made explicit. Although the change is small, this 'fresh ideal world' is significantly different from the regular ideal world. We formally show this as an attack between these two worlds in Section 5 using $\mathcal{O}\left(2^{3n/4}\right)$ queries. This attack works in the sequential indifferentiability setting, so even there the simplification of (implicitly) replacing the regular ideal world with the fresh ideal world is not possible when a security level of more than $3n/4$ bits is desired.

Algorithm 2 The 'fresh ideal world'

function $\mathcal{R}'(x)$
  if $x \in \mathsf{domain}(\mathsf{F})$ **then**
    return $\mathsf{F}(x)$
  $z \xleftarrow{\$} \{0,1\}^n$
  if query from distinguisher **then**
    $\mathsf{F}(x) \leftarrow z$
  return z

function $\mathcal{S}'_b[\mathcal{R}'](x)$
  $y_b \leftarrow \mathcal{S}_b[\mathcal{R}'](x)$
  $y_{1-b} \leftarrow \mathcal{S}_{1-b}[\mathcal{R}'](x)$
  $\mathsf{F}(x) \leftarrow y_b \oplus y_{1-b}$
  return y_b

function $\mathcal{S}'^{-1}_b[\mathcal{R}'](y_b)$
  $x \leftarrow \mathcal{S}_b^{-1}[\mathcal{R}'](y_b)$
  $y_{1-b} \leftarrow \mathcal{S}_b[\mathcal{R}'](x)$
  $\mathsf{F}(x) \leftarrow y_b \oplus y_{1-b}$
  return x

3.3 Random Range

The final problem we describe is that the ranges R_0 and R_1 cannot be considered to be random subsets of $\{0,1\}^n$ as by using inverse queries (halve of) these sets can basically be constructed as desired. These sets were actually considered to be random in [20] as highlighted and fixed in [25]. We note that this same problem is actually again present in [7] as their Lemma 1 considers these sets to be random and independent subsets of $\{0,1\}^n$. This is a fundamental problem in the proof and cannot be salvaged, bare the disallowance of any inverse queries.

4 New (Regular) Indifferentiability Proof

In this section we show that the F construction is regularly indifferentiable from a random oracle $\mathcal{R}$. Surprisingly, we show that the Mennink-Preneel's simulator [25], identical to the uniform simulator from Definition 5 with $\ell = 2$, suffices to provide security up to $2^{2n/3}/n$ queries.

In what follows, we first give a general security lifting lemma in the context of F. Specifically, we characterize the indifferentiability advantage into three terms: (1) the sequential indifferentiability advantage, (2) the failure probability and (3) the sequential difference. As the proof in [25] does hold in the sequential indifferentiability setting, we can directly use this result for term (1). This work also implicitly bounds the failure probability, but, again, only in the sequential setting. We show an upper bound for (2) that also holds in the regular indifferentiability setting by using straightforward computations. The sequential difference is more involved and highly non-trivial. It is the part that is ignored in previous papers. Our approach is generic with the potential of allowing similar resolutions for other works with the same issue. We argue that we can swap two consecutive primitive queries with only receiving a loss of $\mathcal{O}\left(q/2^{2n}\right)$. We use this fact to move the primitive queries corresponding to construction queries around. Instead of having them at the end of the interaction, we move them to just before their construction companions. By doing this, we are able to put the construction queries at the end, as we already know their output, making the transcript suitable for the sequential setting. As there are most q such queries that we have to swap at most q times, this results in an extra $\mathcal{O}\left(q^3/2^{2n}\right)$ term for (3), maintaining the proven security level.

4.1 Assumptions

In order to simplify our analysis, we make the following assumptions on the indifferentiability game, none of which result in a loss of generality for our subsequent derivation of an upper bound for the indifferentiability advantage (b will always denote an unspecified bit):

– The simulator never aborts on a forward primitive query.

– Every forward primitive query x is broadcast to both primitives immediately, and the adversary receives both $y_0 = \mathcal{P}_0(x)$ and $y_1 = \mathcal{P}_1(x)$ as response; as such we think of a forward query as just an input x without specifying one of the two primitives.
– After every inverse primitive query y_b to $\mathcal{P}_b^{-1}$, the response x is immediately fed to the other primitive, and the adversary receives both x and $y_{1-b} = \mathcal{P}_{1-b}(x)$ (the only exception being when $x = \perp$); this along with the first assumption means that every primitive query can be represented in the transcript as a triple (x, y_0, y_1).
– For every construction query x made by the adversary, the transcript contains a corresponding primitive query-response triple (x, y_0, y_1) for some y_0, y_1; this can be enforced by processing all the missing x's as forward primitive queries at the end of the game and appending the generated triples at the end of the transcript.
– Every primitive query-response triple (x, y_0, y_1) in the transcript corresponds to some construction query x; this can similarly be enforced by querying all the missing x's to the construction (or random oracle) at the end of the game.
– The adversary never makes a repeated construction query or a *pointless* primitive query (i.e., a primitive query which has already been settled while processing a previous primitive query).

A construction query-response pair (x, z) and a primitive query-response triple (x, y_0, y_1) which share the same x will be called *companion queries*; the above assumptions then imply that the queries in the transcript always occur in such companion pairs. When needed, we will write a primitive query-response triple as $(x, y_0, y_1)^+$ and $(x, y_0, y_1)_b^-$ to denote a query to $\mathcal{S}^+$ and $\mathcal{S}_b^-$ respectively. Aborted queries to $\mathcal{S}_b^-$ will be denoted as $(y_b, \perp)_b^-$ or simply as $(y_b, \perp)$ when b is either irrelevant or clear from the context.

4.2 Transcript

A transcript τ of length σ is a sequence $(\tau_1, \ldots, \tau_\sigma)$, where for all $j \in [\sigma]$, τ_j is one of the following:

– a construction query-response pair (x, z);
– a completed primitive query-response triple (x, y_0, y_1);
– an aborted primitive query $(y_b, \perp)$.

Accordingly, we partition $[\sigma]$ into three sets $\mathscr{c}(\tau)$, $\mathscr{p}(\tau)$, and $\mathscr{a}(\tau)$, which correspond respectively to the three cases for τ_j above.

Definition 6 (Sequential Transcript). *A transcript τ is said to be sequential when all the construction queries are at the end, i.e., when $\mathscr{c}(\tau)$ exactly coincides with $[i..\sigma]$ for some i.*

We can transform each transcript $\tau = (\tau_1, \ldots, \tau_\sigma)$ to a sequential transcript through the following steps:

1. for each $i \in \mathscr{c}(\tau)$, look at the companion $i' \in \mathscr{p}(\tau)$ (i.e., the i' such that τ_i and $\tau_{i'}$ share the same x), and if $i < i'$, put $\tau_{i'}$ at position i while pushing each of $\tau_i, \ldots, \tau_{i'-1}$ one place to the right;
2. once all construction queries are to the right of their companion primitive queries, push all the construction queries to the end of the transcript without changing their order.

We denote by $\widehat{\tau}$ the output of the above transformation on τ, and treat $\tau \mapsto \widehat{\tau}$ as a mapping from transcripts to sequential transcripts. Algorithm 3 gives an algorithmic description of this transformation. We use the array-indexing notation $\widehat{\tau}[i]$ to indicate the current i-th element of the array $\widehat{\tau}$, to emphasise the dynamic nature of $\widehat{\tau}$ while the algorithm is running.

Algorithm 3 Sequentialising a transcript

```
1: function SEQ(τ)
2:     q ← |τ|
3:     τ̂ ← τ
4:     for i ← 1 to q − 1 do                                   ▷ Step 1
5:         if τ̂[i] is a construction query then
6:             (x, _) ← τ̂[i]
7:             for i′ ← i + 1 to q do                          ▷ Companion search
8:                 if τ̂[i′] is a primitive query then
9:                     (x′, _, _) ← τ̂[i′]
10:                    if x = x′ then                          ▷ Companion detection
11:                        temp ← τ̂[i′]
12:                        for j ← 1 to i′ − i do
13:                            τ̂[i′ − j + 1] ← τ̂[i′ − j]
                           τ̂[i] ← temp
14:    for i ← 1 to q − 1 do                                   ▷ Step 2
15:        if τ̂[i] is a construction query then
16:            for i′ ← i + 1 to q do
17:                if τ̂[i′] is a primitive query then
18:                    temp ← τ̂[i′]
19:                    for j ← 1 to i′ − i do
20:                        τ̂[i′ − j + 1] ← τ̂[i′ − j]
                       τ̂[i] ← temp
21:    return τ̂
```

4.3 Additional Notation

Any adversary $\mathcal{D}$ can be viewed as a two-stage algorithm $(\mathcal{D}_{\mathtt{int}}, \mathcal{D}_{\mathtt{dist}})$, where $\mathcal{D}_{\mathtt{int}}$ and $\mathcal{D}_{\mathtt{dist}}$ represent $\mathcal{D}$'s interactive and distinguishing phases, respectively. Formally, $\mathcal{D}_{\mathtt{int}}$ is an interactive oracle algorithm that outputs a transcript of its interaction with its oracle, and $\mathcal{D}_{\mathtt{dist}}$ is an algorithm that takes as input the

transcript generated by $\mathcal{D}_{\mathtt{int}}$'s interaction with its oracle and outputs a guess bit.

Fix q and a simulator $\mathcal{S}$. Let $\mathcal{T}^{\mathcal{D}}$ denote the set of all possible transcripts consisting of exactly q construction queries and q primitive queries (in companion pairs) that can be realized by $\mathcal{D}$ in an interaction with $(\mathcal{R}, \mathcal{S})$ (we call such a game a (q,q)-query game). For all $\tau \in \mathcal{T}$, we write $\mathbb{P}_{\mathrm{re}}[\tau]$ and $\mathbb{P}_{\mathrm{id}}[\tau]$ to denote the probability of realizing τ by an interaction with (F, Π) (the real world) and $(\mathcal{R}, \mathcal{S})$ (the ideal world) respectively; since we are considering only deterministic adversaries, this probability only depends on the random coins of the oracles and not on $\mathcal{D}$. By extending this notation, we write $\mathbb{P}_{\mathrm{re}}[\mathcal{T}']$ to denote $\sum_{\tau \in \mathcal{T}'} \mathbb{P}_{\mathrm{re}}[\tau]$ for any $\mathcal{T}' \subseteq \mathcal{T}^{\mathcal{D}}$. More generally, for an event $\mathtt{E}$, we also use $\mathbb{P}_{\mathrm{re}}[\mathtt{E}]$ and $\mathbb{P}_{\mathrm{id}}[\mathtt{E}]$ to denote the probability of $\mathtt{E}$ in the real world and the ideal world respectively. Let

- $\mathcal{T}^{\mathcal{D}\hookrightarrow 1} := \left\{\tau \in \mathcal{T}^{\mathcal{D}} \mid \mathcal{D}_{\mathtt{dist}}(\tau) = 1\right\}$;
- $\mathcal{T}^{\mathcal{D}\hookrightarrow 0} := \left\{\tau \in \mathcal{T}^{\mathcal{D}} \mid \mathcal{D}_{\mathtt{dist}}(\tau) = 0\right\} = \mathcal{T}^{\mathcal{D}} \setminus \mathcal{T}^{\mathcal{D}\hookrightarrow 1}$;
- $\mathcal{T}^{\mathcal{D}\geqslant} := \left\{\tau \in \mathcal{T}^{\mathcal{D}} \mid \mathbb{P}_{\mathrm{id}}[\tau] \geqslant \mathbb{P}_{\mathrm{re}}[\tau]\right\}$;
- $\mathcal{T}^{\mathcal{D}}_{\mathrm{bad}} := \{\tau \in \mathcal{T}^{\mathcal{D}} \mid \mathscr{a}(\tau) \neq \varnothing\}$;
- $\mathcal{T}^{\mathcal{D}}_{\mathrm{good}} := \{\tau \in \mathcal{T}^{\mathcal{D}} \mid \mathscr{a}(\tau) = \varnothing\} = \mathcal{T}^{\mathcal{D}} \setminus \mathcal{T}^{\mathcal{D}}_{\mathrm{bad}}$.

For brevity, we also let $\mathcal{T}^{\mathcal{D}\geqslant}_{\mathrm{good}} = \mathcal{T}^{\mathcal{D}\geqslant} \cap \mathcal{T}^{\mathcal{D}}_{\mathrm{good}}$ and $\mathcal{T}^{\mathcal{D}\geqslant}_{\mathrm{bad}} = \mathcal{T}^{\mathcal{D}\geqslant} \cap \mathcal{T}^{\mathcal{D}}_{\mathrm{bad}}$.

Remark 2. We note that $\mathcal{T}^{\mathcal{D}}$ and its various subsets defined above depend on q and $\mathcal{S}$, and $\mathbb{P}_{\mathrm{id}}[\tau]$ also depends on $\mathcal{S}$; when we need to make this dependence explicit, we will add the relevant symbols to the notation—for instance, $\mathcal{T}^{\mathcal{D}}$ becomes $\mathcal{T}^{\mathcal{D}}(q, \mathcal{S})$, and $\mathbb{P}_{\mathrm{id}}[\tau]$ becomes $\mathbb{P}^{\mathcal{S}}_{\mathrm{id}}[\tau]$. Fortunately, most often q and $\mathcal{S}$ will be clear from context, allowing us to drop these explicit references and keep the notation cleaner. We always assume that the adversary $\mathcal{D}$ adapts to q; this could for instance be realised by letting $\mathcal{D}$ be a collection of several instances $\mathcal{D}^q$ for different choices of q, such that $\mathcal{D}^q$ is specifically tailored for playing a (q,q)-query game.

Definition 7 (Failure Probability). *We define the failure probability of $\mathcal{S}$ in a (q,q)-query game as*

$$\mathsf{FP}(q, \mathcal{S}) := \max_{\mathcal{D}} \ \mathbb{P}^{\mathcal{S}}_{\mathrm{id}}\left[\mathcal{T}^{\mathcal{D}}_{\mathrm{bad}}(q, \mathcal{S})\right].$$

Definition 8 (Sequential Difference). *We define the sequential difference of $\mathcal{S}$ in a (q,q)-query game as*

$$\mathsf{SD}(q, \mathcal{S}) := \max_{\mathcal{D}} \sum_{\tau \in \mathcal{T}^{\mathcal{D}\geqslant}_{\mathrm{good}}(q, \mathcal{S})} \left(\mathbb{P}^{\mathcal{S}}_{\mathrm{id}}[\tau] - \mathbb{P}^{\mathcal{S}}_{\mathrm{id}}[\widehat{\tau}]\right).$$

In Lemma 2 of the Supplementary Material we show that for any q and any simulator $\mathcal{S}$ we have

$$\mathbf{Adv}^{\mathrm{indif}}_{F,\mathcal{S}}(q,q) \leqslant \mathbf{Adv}^{\text{seq-indif}}_{F,\mathcal{S}}(q,q) + \mathsf{SD}(q, \mathcal{S}) + \mathsf{FP}(q, \mathcal{S}). \tag{3}$$

4.4 Mennink-Preneel Simulator

We aim to employ (3) with the same simulator used in [25] to achieve security up to $2^{2n/3}/n$ queries. This simulator is identical to the uniform simulator from Definition 5 with $\ell = 2$. From now on, we denote this simulator by $\mathcal{S}$.

First we show in Lemma 3 of the Supplementary Material that the simulator has a limited probability to fail. That is, for the simulator $\mathcal{S}$ we have

$$\mathsf{FP}\left(q, \mathcal{S}\right) \leqslant \frac{2q}{2^n} + \frac{13q^3}{2^{2n}}. \tag{4}$$

We continue with the main part of the proof, showing that the sequential difference of $\mathcal{S}$ is bounded by $\mathcal{O}\left(q^3/2^{2n}\right)$.

Lemma 1. *For Mennink-Preneel's simulator $\mathcal{S}$, we have*

$$\mathsf{SD}\left(q, \mathcal{S}\right) \leqslant \frac{46q^3}{2^{2n}}.$$

Proof. Fix an adversary $\mathcal{D}$. For any $\tau = (\tau_1, \ldots, \tau_{2q}) \in \mathcal{T}_{\text{good}}^{\mathcal{D}\geqslant}$, let C^τ denote the partially sampled $\mathcal{R}$ as revealed to the adversary through all the construction queries in the game, and let $\mathsf{D}_{\mathsf{C}^\tau}$ and $\mathsf{R}_{\mathsf{C}^\tau}$ be respectively the domain and range of C^τ.

For some $\tau = (\tau_1, \ldots, \tau_{2q}) \in \mathcal{T}_{\text{good}}^{\mathcal{D}\geqslant}$, consider primitive queries $\tau_i = (x, y_0, y_1), \tau_j = (x', y_0', y_1')$ with $i < j$, and let R_0 and R_1 respectively denote the range of P_0 and the range of P_1 right before the i-th query. We call the pair (τ_i, τ_j) *erratic* when it satisfies one of the following:

- τ_j is queried to $\mathcal{S}^+$, and $\{\mathsf{C}^\tau(x) \oplus y_1', y_1 \oplus \mathsf{C}^\tau(x')\} \cap \mathsf{R}_0 \neq \varnothing$;
- τ_j is queried to $\mathcal{S}^+$, and $\{\mathsf{C}^\tau(x) \oplus y_0', y_0 \oplus \mathsf{C}^\tau(x')\} \cap \mathsf{R}_1 \neq \varnothing$;
- τ_j is queried to $\mathcal{S}_b^-$, and $y_b \oplus y_{1-b}' \in \mathsf{R}_{\mathsf{C}^\tau}$.

As y_0, y_1, y_0' and y_1' are all sampled uniformly from at most $2^n - 2q$ values and y_{1-b}' is sampled from $2^n - q$ values when $x' \notin \mathsf{D}_{\mathsf{C}^\tau}$, and the event $x' \in \mathsf{D}_{\mathsf{C}^\tau}$ happens with probability at most $q/(2^n - q)$, we can bound the probability by

$$\begin{aligned}\mathbb{P}_{\text{id}}\left[(\tau_i, \tau_j) \text{ erratic}\right] &\leqslant \max\left(\frac{2q}{2^n - 2q} + \frac{2q}{2^n - 2q}, \frac{q}{2^n - q} + \frac{q}{2^n - q}\right) \\ &\leqslant \max\left(\frac{4q}{2^n} + \frac{4q}{2^n}, \frac{2q}{2^n} + \frac{2q}{2^n}\right) = \frac{8q}{2^n}, \end{aligned} \tag{5}$$

using that $q \leqslant 2^{n-2}$. We partition $\mathcal{T}_{\text{good}}^{\mathcal{D}\geqslant}$ into the set $\mathcal{T}_{\text{ill}}^{\mathcal{D}\geqslant}$ of *ill-behaved* transcripts, and the set $\mathcal{T}_{\text{well}}^{\mathcal{D}\geqslant}$ of *well-behaved* transcripts, based on the following criterion: $\tau \in \mathcal{T}_{\text{ill}}^{\mathcal{D}\geqslant}$ if for some $x, x', x'' \in \mathsf{D}_{\mathsf{C}^\tau}$, $\mathsf{C}^\tau(x) = \mathsf{C}^\tau(x') = \mathsf{C}^\tau(x'')$. We have that

$$\sum_{\tau \in \mathcal{T}_{\text{ill}}^{\mathcal{D}\geqslant}} \mathbb{P}_{\text{id}}\left[\tau\right] \leqslant \frac{q^3}{2^{2n}}. \tag{6}$$

Fix a transcript $\tau \in \mathcal{T}_{\text{well}}^{\mathcal{D}\geqslant}$, and let $\mathsf{C} := \mathsf{C}^{\tau}$. We first observe that responses obtained from the random oracles are sampled independent of the rest of the game, and $\mathcal{D}$ (eventually) sees the random oracle outputs of all x that occur in primitive query-response triples, so we can always condition on the outputs of all the construction queries when computing the probabilities of simulator responses. In the analysis that follows we assume that all the probabilities are implicitly conditioned on the random oracle outputs (which is the same as assuming that the random oracle output $\mathsf{C}(x)$ is known for each triple (x, y_0, y_1)).

We will find it useful to derive expressions for $\mathbb{P}_{\text{id}}[\tau_i \mid \tau_1, \ldots, \tau_{i-1}]$ for any $i \in \mathcal{p}(\tau)$. Fix i, and define $\tau_{\mathtt{head}} := (\tau_1, \ldots, \tau_{i-1})$. First let $\tau_i = (x, y_0, y_1)^{+}$. Let R_0 and R_1 be the ranges of the partial permutations P_0 and P_1 respectively just before the i-th query. Writing $z := \mathsf{C}(x)$, we have

$$\mathbb{P}_{\text{id}}[\tau_i \mid \tau_{\mathtt{head}}] = \frac{1}{2^n - |\mathsf{R}_0 \cup (\mathsf{R}_1 \oplus z)|}. \tag{7}$$

Next let $\tau_i = (x, y_0, y_1)_0^{-}$. For arbitrary $\mathsf{G} \subseteq \mathsf{D}_{\mathsf{C}}$ and $\mathsf{H} \subseteq \{0,1\}^n$ define

$$\mathsf{S}_{\mathsf{G}\to\mathsf{H}} := \{x \in \mathsf{G} \mid \mathsf{C}(x) \in \mathsf{H} \cap \mathsf{R}_{\mathsf{C}}\},$$

the set of elements in G which have an image under C in H. Finally, let $\mathsf{D} \subseteq \mathsf{D}_{\mathsf{C}}$ be the shared domain of P_0 and P_1 just before the i-th query. (Note that all these sets are functions of $\tau_{\mathtt{head}}$.) In Lemma 4 of the Supplementary Material we show that

$$\mathbb{P}_{\text{id}}[\tau_i \mid \tau_{\mathtt{head}}] = \frac{\left|\mathsf{S}_{\mathsf{D}_{\mathsf{C}}\setminus\mathsf{D}\to\mathsf{R}_1\oplus y_0}\right| + 2^n - q|\mathsf{D}|/2^n}{(2^n - |\mathsf{D}|)^2}. \tag{8}$$

We are now ready to derive the bound claimed in the lemma statement. Define

$$\Delta_{\tau} := \mathbb{P}_{\text{id}}[\tau] - \mathbb{P}_{\text{id}}[\widehat{\tau}].$$

For each construction-query index $i \in \mathcal{c}(\tau)$, let i^* denote the companion primitive-query index in $\mathcal{p}(\tau)$ (i.e., $\tau_i = (x, z)$ and $\tau_{i^*} = (x, y_0, y_1)$ for some x, y_0, y_1, z). We recall that we construct $\widehat{\tau}$ from τ as follows:

- for each $i \in \mathcal{c}(\tau)$, if $i < i^*$ then put τ_{i^*} at position i while pushing each of $\tau_i, \ldots, \tau_{i^*-1}$ one place to the right.
- once all construction queries are to the right of their companion primitive queries, push all the construction queries to the end.

We first observe that the second step above does not affect the probability of the transcript, because the output of the construction queries is already fixed from the companion primitive queries. (Note that this may not hold for certain simulators which try to cheat by returning a query-response triple (x, y_0, y_1) without ensuring that $y_0 \oplus y_1 = \mathcal{R}(x)$, but it holds for our simulator.) Thus, when computing $\mathbb{P}_{\text{id}}[\widehat{\tau}]$ we can pretend that the second step did not happen.

In the first step, we can assume that we check each $i \in \mathcal{c}(\tau)$ in order. Consider the smallest $i \in \mathcal{c}(\tau)$ satisfying $i < i^*$. Then, τ_{i^*} moves $i^* - i$ places to the left.

This can be seen as a sequence of $i^* - i$ adjacent transpositions, where the j-th transposition consists of swapping τ_{i^*} with τ_{i^*-j}. Let $\tau^{(i,j)}$ denote the transcript obtained after j transpositions, with the convention that $\tau^{(i,0)} = \tau$; also let $\tau^{[i]} := \tau^{(i,i^*-i)}$ denote the transcript at the moment when τ_{i^*} has reached the target position. We also add i to a (mutable) set $\mathcal{I}$, initialised as empty, which will eventually hold all the indices which need to be *processed* as above through a sequence of adjacent transpositions.

We can inductively extend this notation for the rest of the transpositions as follows: having obtained $\tau^{[i']}$ for the latest (and largest) $i' \in \mathcal{I}$, we look for the smallest $i \in \mathscr{c}\left(\tau^{[i']}\right)$ satisfying $i < i^*$; we then move $\tau_{i^*}^{[i']}$ $i^* - i$ places to the left through $i^* - i$ adjacent transpositions, the j-th of which swaps $\tau_{i^*}^{[i']}$ with $\tau_{i^*-j}^{[i']}$; and finally, i is added to $\mathcal{I}$. $\tau^{(i,j)}$ continues to denote the transcript obtained after j transpositions, with the convention that $\tau^{(i,0)} = \tau^{[i']}$, and $\tau^{[i]} := \tau^{(i,i^*-i)}$ now denotes the transcript at the moment when $\tau_{i^*}^{[i']}$ has reached the target position. For the last i to be added to $\mathcal{I}$, $\tau^{[i]} = \widehat{\tau}$.

Remark 3. We point out that it would be difficult to define the notation by listing out at the outset all the $i \in \mathscr{c}(\tau)$ satisfying $i < i^*$ and going through them one by one; this is because processing the i-th entry changes the positions of the next $i^* - i$ entries, which could contain the next candidate to be processed. We further point out that the above notation is nevertheless well-defined, because the positions of the candidate entries can only increase during the handling of previous candidates, and we process them in increasing order, thus ensuring the same i is never repeated.

Remark 4. One part of the notation we abuse is i^*, which we assume is defined at every stage in accordance with the transcript $\tau^{(i,0)} = \tau^{[i']}$ for the immediate predecessor i' of i in $\mathcal{I}$, i.e., it shows the position of a query in $\mathscr{p}\left(\tau^{[i']}\right)$. Making its dependence on τ explicit would make the notation more cumbersome, so we deem it best for the sake of clarity to leave this dependence implicit and add this clarifying remark. It may be worth noting that the condition $i < i^*$ is invariant under the processing of previous entries, even with the lazy definition of i^*.

For an $i \in \mathcal{I}$ and a $j \in [i^* - i]$, we observe that the first $i^* - j - 1$ entries are identical in $\tau^{(i,j-1)}$ and $\tau^{(i,j)}$; we call this common prefix $\tau_{\mathsf{head}}^{(i,j)}$. Similarly, the last $\sigma - i^* + j - 1$ entries of the transcripts are also identical, forming a common suffix we call $\tau_{\mathsf{tail}}^{(i,j)}$. Then we see that

$$\begin{aligned}
\mathbb{P}_{\mathrm{id}}\left[\tau^{(i,j-1)}\right] &= \mathbb{P}_{\mathrm{id}}\left[\tau_{\mathsf{head}}^{(i,j)}\right] \cdot \mathbb{P}_{\mathrm{id}}\left[\tau_{i^*-j} \mid \tau_{\mathsf{head}}^{(i,j)}\right] \\
&\qquad \cdot \mathbb{P}_{\mathrm{id}}\left[\tau_{i^*} \mid \tau_{i^*-j}, \tau_{\mathsf{head}}^{(i,j)}\right] \cdot \mathbb{P}_{\mathrm{id}}\left[\tau_{\mathsf{tail}}^{(i,j)} \mid \tau_{i^*}, \tau_{i^*-j}, \tau_{\mathsf{head}}^{(i,j)}\right], \\
\mathbb{P}_{\mathrm{id}}\left[\tau^{(i,j)}\right] &= \mathbb{P}_{\mathrm{id}}\left[\tau_{\mathsf{head}}^{(i,j)}\right] \cdot \mathbb{P}_{\mathrm{id}}\left[\tau_{i^*} \mid \tau_{\mathsf{head}}^{(i,j)}\right] \\
&\qquad \cdot \mathbb{P}_{\mathrm{id}}\left[\tau_{i^*-j} \mid \tau_{i^*}, \tau_{\mathsf{head}}^{(i,j)}\right] \cdot \mathbb{P}_{\mathrm{id}}\left[\tau_{\mathsf{tail}}^{(i,j)} \mid \tau_{i^*-j}, \tau_{i^*}, \tau_{\mathsf{head}}^{(i,j)}\right].
\end{aligned}$$

We define

$$\rho_\tau^{(i,j)} := \frac{\mathbb{P}_{\mathrm{id}}\left[\tau^{(i,j-1)}\right]}{\mathbb{P}_{\mathrm{id}}\left[\tau^{(i,j)}\right]} = \frac{\mathbb{P}_{\mathrm{id}}\left[\tau_{i^*-j} \mid \tau_{\mathsf{head}}^{(i,j)}\right] \cdot \mathbb{P}_{\mathrm{id}}\left[\tau_{i^*} \mid \tau_{i^*-j}, \tau_{\mathsf{head}}^{(i,j)}\right]}{\mathbb{P}_{\mathrm{id}}\left[\tau_{i^*} \mid \tau_{\mathsf{head}}^{(i,j)}\right] \cdot \mathbb{P}_{\mathrm{id}}\left[\tau_{i^*-j} \mid \tau_{i^*}, \tau_{\mathsf{head}}^{(i,j)}\right]}. \tag{9}$$

For each $i \in \mathcal{I}$, we further define

$$\rho_\tau^{[i]} := \prod_{j \in [i^*-i]} \rho_\tau^{(i,j)} = \frac{\mathbb{P}_{\mathrm{id}}\left[\tau^{(i,0)}\right]}{\mathbb{P}_{\mathrm{id}}\left[\tau^{[i]}\right]},$$

and finally, we define

$$\rho_\tau := \prod_{i \in \mathcal{I}} \rho_\tau^{[i]} = \frac{\mathbb{P}_{\mathrm{id}}\left[\tau\right]}{\mathbb{P}_{\mathrm{id}}\left[\widehat{\tau}\right]}.$$

Using this, and the fact that $1/(1+x) \geqslant 1-x$ for all x, we can write

$$\begin{aligned}
\Delta_\tau = \left(1 - \frac{1}{\rho_\tau}\right) \cdot \mathbb{P}_{\mathrm{id}}\left[\tau\right] &= \left(1 - \prod_{i \in \mathcal{I}} \prod_{j \in [i^*-i]} \frac{1}{\rho_\tau^{(i,j)}}\right) \cdot \mathbb{P}_{\mathrm{id}}\left[\tau\right] \\
&\leqslant \left(1 - \prod_{i \in \mathcal{I}} \prod_{j \in [i^*-i]} \left(1 - (\rho_\tau^{(i,j)} - 1)\right)\right) \cdot \mathbb{P}_{\mathrm{id}}\left[\tau\right] \\
&\leqslant \sum_{i \in \mathcal{I}} \sum_{j \in [i^*-i]} (\rho_\tau^{(i,j)} - 1) \cdot \mathbb{P}_{\mathrm{id}}\left[\tau\right].
\end{aligned}$$

We next try to to find a suitable upper bound for ρ_τ. Fix an $i \in \mathcal{I}$ and a $j \in [i^* - i]$. Our first task will be to find an upper bound for $\rho_\tau^{(i,j)}$. In Lemma 5 of the Supplementary Material we find one depending on whether $(\tau_{i^*-j}, \tau_{i^*})$ is an erratic pair or not. We get

$$\rho_\tau^{(i,j)} \leqslant \Phi^{(i,j)}(\tau) := \begin{cases} 1 + \dfrac{5}{2^{2n}} & \text{if } (\tau_{i^*-j}, \tau_{i^*}) \text{ is erratic,} \\ 1 + \dfrac{5q}{2^{2n}} & \text{otherwise,} \end{cases}$$

for all $\tau \in \mathcal{T}_{\mathrm{well}}^{\mathcal{D}\geqslant}$. Furthermore, as the probability that $(\tau_{i^*-j}, \tau_{i^*})$ is an erratic pair is at most $8q/2^n$ by (5) we derive for $\Phi^{(i,j)}$ that

$$\begin{aligned}
\mathbb{E}_{\mathrm{id}}\left[\Phi^{(i,j)}(\tau)\right] &\leqslant \left(1 + \frac{5}{2^n}\right) \mathbb{P}_{\mathrm{id}}\left[(\tau_i, \tau_j) \text{ erratic}\right] \\
&\quad + \left(1 + \frac{5q}{2^{2n}}\right) \mathbb{P}_{\mathrm{id}}\left[(\tau_i, \tau_j) \text{ not erratic}\right] \\
&\leqslant 1 + \frac{40q}{2^{2n}} + \frac{5q}{2^{2n}} = 1 + \frac{45q}{2^{2n}},
\end{aligned}$$

Using this bound and the one in (6), and extending the definition of Δ_τ to all of $\mathcal{T}_{\text{good}}^{\mathcal{D}\geqslant}$, we have

$$\begin{aligned}
\mathsf{SD}\left(q,\mathcal{S}\right) &= \max_{\mathcal{D}} \sum_{\tau\in\mathcal{T}_{\text{good}}^{\mathcal{D}\geqslant}} \Delta_\tau = \max_{\mathcal{D}} \left(\sum_{\tau\in\mathcal{T}_{\text{well}}^{\mathcal{D}\geqslant}} \Delta_\tau + \sum_{\tau\in\mathcal{T}_{\text{ill}}^{\mathcal{D}\geqslant}} \Delta_\tau \right) \\
&\leqslant \max_{\mathcal{D}} \left(\sum_{\tau\in\mathcal{T}_{\text{well}}^{\mathcal{D}\geqslant}} \sum_{i\in\mathcal{I}} \sum_{j\in[i^*-i]} (\rho_\tau^{(i,j)} - 1)\cdot \mathbb{P}_{\text{id}}\left[\tau\right] + \sum_{\tau\in\mathcal{T}_{\text{ill}}^{\mathcal{D}\geqslant}} \mathbb{P}_{\text{id}}\left[\tau\right] \right) \\
&\leqslant \max_{\mathcal{D}} \left(\sum_{i\in\mathcal{I}} \sum_{j\in[i^*-i]} \mathbb{E}_{\text{id}}\left[\Phi^{(i,j)}(\tau) - 1\right] + \sum_{\tau\in\mathcal{T}_{\text{ill}}^{\mathcal{D}\geqslant}} \mathbb{P}_{\text{id}}\left[\tau\right] \right) \\
&\leqslant \max_{\mathcal{D}} \left(\frac{45q^3}{2^{2n}} + \frac{q^3}{2^{2n}} \right) = \frac{46q^3}{2^{2n}},
\end{aligned}$$

thus establishing Lemma 1. □

Using equations (3), (4) and Lemma 1, and the observation from [15] that [25, Theorem 2] implies sequential indifferentiability with the same advantage, we get the following full indifferentiability bound.

Corollary 1. *For $9n \leqslant q \leqslant 2^{n-2}$, there exists a simulator $\mathcal{S}$ making at most $2q$ queries to $\mathcal{R}$ such that*

$$\mathbf{Adv}_{F,\mathcal{S}}^{\text{indif}}(q,q) \leqslant \sqrt{\frac{9nq^3}{2^{2n}}} + \frac{2q}{2^n} + \frac{59q^3}{2^{2n}}.$$

5 Sequential Difference 'Fresh Ideal World'

In this section, we show an attack with complexity $\mathcal{O}\left(2^{3n/4}\right)$ that can distinguish the ideal world from the 'fresh ideal world' as described in Section 3.2. It works for any uniform simulator as described in Definition 5. Moreover, the attack makes primitive queries before making construction queries, meaning that the simplification of considering random oracle outputs fresh is even problematic when considering the weaker sequential indifferentiability setting.

The intuition behind this attack is that in the regular ideal world the construction oracle output $\mathcal{C}(x)$ is uniformly at random and independently selected from $\{0,1\}^n$ at start and does not change during the interaction. However, in the fresh ideal world the output $\mathcal{C}(x)$ is not fixed at the start but is redrawn every time. Consequently, some biases introduced when making well-tailored backward queries, where the simulator does not select values with specific construction oracle outputs, are not present. We can exploit this flaw to differentiate between these worlds using $\mathcal{O}\left(2^{3n/4}\right)$ queries as shown below.

5.1 Attack Setup

Let $q = 2^k$ for some k with $q \leqslant 2^{n-1}$. We make at most $2q$ queries to the primitive oracle and at most q queries to the construction oracle. We define the distinguisher $\mathcal{D}$ as follows:

1. Let $X \subseteq \{0,1\}^n$ be an arbitrary set of size q;
2. Call $\mathcal{P}_0^{-1}(0^{n-k} \parallel y)$ for all $y \in \{0,1\}^k$;
3. Call $\mathcal{P}_1^{-1}(0^{n-k} \parallel y)$ for all $y \in \{0,1\}^k$;
4. Call $\mathcal{C}(x)$ for all $x \in X$;
5. Count the number of $x \in X$ such that $\lfloor \mathcal{C}(x) \rfloor_{n-k} = 0^{n-k}$ and call it c;
6. Return 1 when c is lower than some cutoff d and 0 otherwise.

Note that as pointless queries are not made, the value of $\mathcal{C}(x)$ can be determined in either step (2), (3) or (4). The cutoff d is the midpoint between the expected values in the two different worlds. The distinguisher $\mathcal{D}$ is formally given in Algorithm 4, with implicit calls by the simulator made explicit.

We will compute the expectation and variance of c in both the regular ideal world (μ_1 and σ_1^2) and the fresh ideal world (μ_2 and σ_2^2). These values will be used to determine the advantage.

5.2 Ideal World

In the regular ideal world, we can view $\mathcal{C}(x)$ to be fixed at the start for all x. It does not matter whether the attacker retrieves it in step (2), (3) or (4), the probability that $\lfloor \mathcal{C}(x) \rfloor_{n-k} = 0^{n-k}$ for a fixed $x \in X$ is always $1/2^{n-k} = q/2^n$. As every x is also independent from the other values, the total count is distributed as the binomial distribution $B(q, q/2^n)$, leading to

$$\mu_1 = \frac{q^2}{2^n},$$
$$\sigma_1^2 = \frac{q^2}{2^n}\left(1 - \frac{q}{2^n}\right) \leqslant \frac{q^2}{2^n}.$$

5.3 Fresh Ideal World

In the fresh ideal world, however, $\mathcal{C}(x)$ is not fixed at the start but is sampled fresh at every invocation. This means that we have to separate the different steps. Let $c_2^{(2)}$, $c_2^{(3)}$ and $c_2^{(4)}$ denote the subcounts in step (2), (3) and (4), respectively, so that $c_2 = c_2^{(2)} + c_2^{(3)} + c_2^{(4)}$. We compute them separately.

For $c_2^{(2)}$ we consider the output of an arbitrary query made in step (2). As the simulator samples x uniformly from all possibilities, the probability that $x \in X$ is $q/2^n$. Furthermore, the simulator samples $\mathcal{C}(x)$ uniformly over all possibilities such that $\mathcal{C}(x) \notin R_1 \oplus (0^{n-k} \parallel y)$. As the previous queries are also sampled fresh, R_1 is uniformly distributed over all possible subsets, hence there is no bias

Algorithm 4 Distinguisher between the normal and 'fresh ideal world'

function $\mathcal{D}^{\mathcal{P},\mathcal{C}}$
  $X \leftarrow \{\, 0^{n-k} \parallel x : x \in \{0,1\}^k \,\}$
  $c^{(2)} \leftarrow$ CountInverse(0)
  $c^{(3)} \leftarrow$ CountInverse(1)
  $c^{(4)} \leftarrow$ CountConstruction()
  $c \leftarrow c^{(2)} + c^{(3)} + c^{(4)}$
  if $c \leqslant d$ **then**
    return 1
  else
    return 0

function CountInverse(b)
  $c \leftarrow 0$
  for $y \in \{0,1\}^k$ **do**
    $y_b \leftarrow 0^{n-k} \parallel y$
    if $y_b \notin \mathsf{range}(\mathcal{P}_b)$ **then**
      $x \leftarrow \mathcal{P}_b^{-1}(y_b)$
      $y_{1-b} \leftarrow \mathcal{P}_{1-b}(x)$
      $z \leftarrow y_b \oplus y_{1-b}$
      if $x \in X$ and $\lfloor z \rfloor_{n-k} = 0^{n-k}$ **then**
        $c \leftarrow c + 1$
  return c

function CountConstruction
  $c \leftarrow 0$
  for $x \in X$ **do**
    if $x \notin \mathsf{domain}(\mathcal{P}_0)$ **then**
      $z \leftarrow \mathcal{C}(x)$
      if $\lfloor z \rfloor_{n-k} = 0^{n-k}$ **then**
        $c \leftarrow c + 1$
  return c

leading to $q/2^n$ for the probability that $\lfloor \mathcal{C}(x) \rfloor_{n-k} = 0^{n-k}$ happens. Finally, as there are always q queries made in step (2), we get

$$\mathbb{E}\left[c_2^{(2)}\right] = q \cdot \frac{q}{2^n} \cdot \frac{q}{2^n} = \frac{q^3}{2^{2n}}.$$

For $c_2^{(3)}$ we simply have that $\lfloor \mathcal{C}(x) \rfloor_{n-k} = 0^{n-k}$ is not possible, as the simulator rejects any x with $\mathcal{C}(x) \in R_0 \oplus (0^{n-k} \parallel y')$ and $\{\, 0^{n-k} \parallel y : y \in \{0,1\}^k \,\} \subseteq R_0$. As a consequence,

$$c_2^{(3)} = 0.$$

Note that the number of queries made in step (3) is not always q as there can pointless queries that are already made in step (2). This happens whenever the simulator sets $\mathcal{P}_1(x)$ as $0^{n-k} \parallel y'$ for a $y' \in \{0,1\}^k$, which happens exactly when $\lfloor \mathcal{C}(x) \rfloor_{n-k} = 0^{n-k}$ which has a probability of $q/2^n$. Therefore, the number of queries made in step (3) has an expected value of $q - q^2/2^n$.

For $c_2^{(4)}$ we have that $\mathcal{C}(x)$ is freshly sampled, hence the probability that $\lfloor \mathcal{C}(x) \rfloor_{n-k} = 0^{n-k}$ happens is $q/2^n$. We still have no bias in the chosen x, so the probability that $x \in X$ for a specific x is also still $q/2^n$. However, we do not necessarily make the maximum possible number of q queries as we ignore pointless queries. Let $q^{(2)}$ and $q^{(3)}$ denote the number of queries made in step (2) and (3), respectively, of which the output lies within X. Then, we have that

$$\mathbb{E}\left[c_2^{(4)} \;\middle|\; q^{(2)}, q^{(3)}\right] = (q - q^{(2)} - q^{(3)})\frac{q}{2^n}.$$

As $\mathbb{E}\left[q^{(2)}\right] = q^2/2^n$ and $\mathbb{E}\left[q^{(3)}\right] = (q - q^2/2^n)q/2^n = q^2/2^n - q^3/2^{2n}$, we get by the law of total expectation that

$$\begin{aligned}\mathbb{E}\left[c_2^{(4)}\right] &= \left(q - \mathbb{E}\left[q^{(2)}\right] - \mathbb{E}\left[q^{(3)}\right]\right)\frac{q}{2^n} = \left(q - \frac{2q^2}{2^n} + \frac{q^3}{2^{2n}}\right)\frac{q}{2^n} \\ &= \frac{q^2}{2^n} - \frac{2q^3}{2^{2n}} + \frac{q^4}{2^{3n}}.\end{aligned}$$

Combining all this gives

$$\begin{aligned}\mu_2 = \mathbb{E}\left[c_2^{(2)} + c_2^{(3)} + c_2^{(4)}\right] &= \frac{q^3}{2^{2n}} + 0 + \frac{q^2}{2^n} - \frac{2q^3}{2^{2n}} + \frac{q^4}{2^{3n}} \\ &= \frac{q^2}{2^n} - \frac{q^3}{2^{2n}} + \frac{q^4}{2^{3n}} \leqslant \frac{q^2}{2^n} - \frac{q^3}{2^{2n+1}},\end{aligned}$$

using that $q \leqslant 2^{n-1}$. For the variance we have that every single query is a Bernoulli variable. In step (2) the probability is $q^2/2^{2n}$ and in step (3) the probability is $q/2^n$, giving variances of $q^2/2^{2n}(1-q^2/2^{2n}) \leqslant q^2/2^{2n}$ and $q/2^n(1-q/2^n) \leqslant q/2^n$, respectively. The variance of a sum of variables is the sum of the variances of the individual variables with the covariances between the variables added. But in our case the variables in step (2) negatively influence future queries and variables in step (3) have no influence, leading to a negative correlation, hence

$$\sigma_2^2 \leqslant \frac{q^3}{2^{2n}} + \frac{q^2}{2^n} \leqslant \frac{2q^2}{2^n},$$

where we additionally use the fact that at most q queries are made in both step (2) and (4).

5.4 Advantage

For the advantage we use Lemma 6 of the Supplementary Material, leading to an advantage of at least

$$\mathbf{Adv}^{\text{seq-indif}}_{(\mathcal{R},\mathcal{S}),(\mathcal{R}',\mathcal{S})}(\mathcal{D}) \geqslant 1 - \frac{4(\sigma_1^2 + \sigma_2^2)}{(\mu_1 - \mu_2)^2} \geqslant 1 - \frac{12q^2}{2^n}\frac{2^{4n+2}}{q^6} \geqslant 1 - \frac{2^{3n+6}}{q^4},$$

where $\mathbf{Adv}^{\text{seq-indif}}_{(\mathcal{R},\mathcal{S}),(\mathcal{R}',\mathcal{S})}(\cdot)$ denotes the sequential indifferentiability advantage between the ideal world $(\mathcal{R}, \mathcal{S})$ and the fresh ideal world $(\mathcal{R}', \mathcal{S})$ with $\mathcal{R}'$ the modified random oracle that always gives fresh results.

6 Generic Differentiability Attack

In this section, we show an attack with complexity $\mathcal{O}\left(2^{5n/6}\right)$ that can distinguish the ideal world with the uniform simulator from the real world. The intuition behind this attack is that the uniform forward simulator returns a value uniformly sampled from all possibilities. While this sounds reasonable it turns out that this actually does not exactly match the real world for all interactions. This uniformity changes the distribution of outputs when the order of the queries is changed in the ideal world, while the order does not matter in the real world. We can exploit this flaw to attack the uniform simulator using $\mathcal{O}\left(2^{5n/6}\right)$ queries as shown below.

6.1 Attack Setup

Let $q = 2^k$ for some k with $q \leqslant 2^{n-3}$. We make at most $3q$ queries to the primitive oracle and at most q queries to the construction oracle.

1. Call $\mathcal{P}_1^{-1}(0^{n-k} \parallel y)$ for all $y \in \{0,1\}^k$;
2. Call $\mathcal{C}(x_i) = z_i$ for q fresh x_i;
3. Let I be the index set consisting of all i such that $\lfloor z_i \rfloor_{n-k} = 0^{n-k}$ in the previous step;
4. Call $\mathcal{P}_0(x_i)$ for all $i \in I$ (optional);
5. Call $\mathcal{P}_0(x_j) = y_j$ for q fresh x_j;
6. Count the number of j such that $\lfloor y_j \rfloor_{n-k} = 0^{n-k}$ and call it c;
7. Return 1 when c is lower than some cutoff d and 0 otherwise.

Step (4) is denoted as optional. We define two related distinguishers depending on whether step (4) is executed or not. We denote $\mathcal{D}_\varnothing$ for the distinguisher that skips (4) and $\mathcal{D}_I$ for the distinguisher that executes (4). Again, the cutoff d is the midpoint between the expected values in the two different worlds. The distinguishers $\mathcal{D}_\varnothing$ and $\mathcal{D}_I$ are formally given in Algorithm 5, with again implicit calls by the simulator made explicit.

By the triangle inequality we get that

$$\begin{aligned}
&\left|\mathbb{P}\left[\mathcal{D}_\varnothing^{\mathcal{R},\mathcal{S},\mathcal{S}^{-1}} = 1\right] - \mathbb{P}\left[\mathcal{D}_I^{\mathcal{R},\mathcal{S},\mathcal{S}^{-1}} = 1\right]\right| \\
&\leqslant \left|\mathbb{P}\left[\mathcal{D}_\varnothing^{\mathcal{R},\mathcal{S},\mathcal{S}^{-1}} = 1\right] - \mathbb{P}\left[\mathcal{D}_\varnothing^{F,\Pi,\Pi^{-1}} = 1\right]\right| \\
&+ \left|\mathbb{P}\left[\mathcal{D}_\varnothing^{F,\Pi,\Pi^{-1}} = 1\right] - \mathbb{P}\left[\mathcal{D}_I^{F,\Pi,\Pi^{-1}} = 1\right]\right| \\
&+ \left|\mathbb{P}\left[\mathcal{D}_I^{F,\Pi,\Pi^{-1}} = 1\right] - \mathbb{P}\left[\mathcal{D}_I^{\mathcal{R},\mathcal{S},\mathcal{S}^{-1}} = 1\right]\right| \\
&= \mathbf{Adv}_{F,\mathcal{S}}^{\text{indif}}(\mathcal{D}_\varnothing) + 0 + \mathbf{Adv}_{F,\mathcal{S}}^{\text{indif}}(\mathcal{D}_I) \\
&\leqslant 2\max\left(\mathbf{Adv}_{F,\mathcal{S}}^{\text{indif}}(\mathcal{D}_\varnothing), \mathbf{Adv}_{F,\mathcal{S}}^{\text{indif}}(\mathcal{D}_I)\right),
\end{aligned} \tag{10}$$

where the reasoning for the 0 is given in Section 6.2. This means that if we show that there is a non-negligible difference between adding the queries I or

Algorithm 5 Distinguisher on the uniform simulator, with the highlighted lines 9–11 included in $\mathcal{D}_I$ but not in $\mathcal{D}_\varnothing$

```
1: function D^{P,C}
2:     for y ∈ {0,1}^k do
3:         y_1 ← 0^{n−k} ‖ y
4:         x ← P_1^{−1}(y_1)
5:         y_0 ← P_0(x)
6:     for 1 ⩽ i ⩽ q do
7:         x ←$ {0,1}^n \ (domain(P_0) ∪ domain(C))
8:         z ← C(x)
9:         if ⌊z⌋_{n−k} = 0^{n−k} then
10:            y_0 ← P_0(x)
11:            y_1 ← P_1(x)
12:    c ← 0
13:    for 1 ⩽ j ⩽ q do
14:        x ←$ {0,1}^n \ (domain(P_0) ∪ domain(C))
15:        y_0 ← P_0(x)
16:        y_1 ← P_1(x)
17:        if ⌊y_0⌋_{n−k} = 0^{n−k} then
18:            c ← c + 1
19:    if c ⩽ d then
20:        return 1
21:    else
22:        return 0
```

not in the ideal world, i.e., there is a non-trivial lower bound for (10), there is a distinguisher (either $\mathcal{D}_\varnothing$ or $\mathcal{D}_I$) that has a non-negligible advantage on the original construction. In Section 6.3 we will derive such a lower bound on (10), leading to

$$2 \max \left(\mathbf{Adv}^{\text{indif}}_{F,\mathcal{S}}(\mathcal{D}_\varnothing), \mathbf{Adv}^{\text{indif}}_{F,\mathcal{S}}(\mathcal{D}_I) \right) \geqslant 1 - \mathcal{O}\left(\frac{2^{5n}}{q^6} \right).$$

6.2 Real World

In the real world, the construction oracle is defined as $F(x) = \Pi_0(x) \oplus \Pi_1(x)$. This means that a construction query $F(x)$ will behave the same as the two primitive queries $\Pi_0(x)$ and $\Pi_1(x)$. Therefore, the primitive queries $\Pi_0(x_i)$ optionally made in step (4) have no influence as they are already implicitly executed in the construction query $F(x_i)$ in step (3). As the only difference between $\mathcal{D}_\varnothing$ and $\mathcal{D}_I$ are these 'extra' primitive queries, their output probabilities do not differ, and hence

$$\mathbb{P}\left[\mathcal{D}_\varnothing^{F,\Pi,\Pi^{-1}} = 1 \right] - \mathbb{P}\left[\mathcal{D}_I^{F,\Pi,\Pi^{-1}} = 1 \right] = 0.$$

6.3 Ideal World

In this section we focus on finding a lower bound for

$$\left|\mathbb{P}\left[\mathcal{D}_{\varnothing}^{\mathcal{R},\mathcal{S},\mathcal{S}^{-1}}=1\right]-\mathbb{P}\left[\mathcal{D}_{I}^{\mathcal{R},\mathcal{S},\mathcal{S}^{-1}}=1\right]\right|.$$

In order to do this, we will compute the expectation and variance of c for both $\mathcal{D}_{\varnothing}$ ($\mu_{\varnothing}$ and $\sigma_{\varnothing}^2$) and $\mathcal{D}_I$ (μ_I and σ_I^2). These values will be used to determine the advantage. We are mostly interested in the difference between the expectations which we can denote as

$$\mu_I-\mu_{\varnothing}=\sum_j \delta_j,$$
$$\delta_j=p_{I,j}-p_{\varnothing,j},$$

where $p_{\varnothing,j}=\mathbb{P}_{\varnothing}\left[\lfloor y_j\rfloor_{n-k}=0^{n-k}\right]$ and $p_{I,j}=\mathbb{P}_I\left[\lfloor y_j\rfloor_{n-k}=0^{n-k}\right]$, where $\mathbb{P}_{\varnothing}[\cdot]$ (resp., $\mathbb{P}_I[\cdot]$) denotes the probability is taken when interacting with distinguisher $\mathcal{D}_{\varnothing}$ (resp., $\mathcal{D}_I$).

Now we will look at the probability that $\lfloor y_j\rfloor_{n-k}=\lfloor\mathcal{S}_0(x_j)\rfloor_{n-k}=0^{n-k}$ for a fixed j when I is excluded or included. By the behavior of the simulator, the probability is of the form

$$p_j=\mathbb{P}\left[\lfloor y_j\rfloor_{n-k}=0^{n-k}\right]=\mathbb{E}\left[\frac{q-W_j}{2^n-V_j}\right],$$

where W_j denotes the number of elements that exclude $\lfloor y_j\rfloor_{n-k}=0^{n-k}$ from occurring and V_j denotes the number of excluded values to draw. This notation is more generic and we denote $W_{\varnothing,j}$ and $V_{\varnothing,j}$ when interacting with $\mathcal{D}_{\varnothing}$ and similar for $\mathcal{D}_I$.

In Lemma 7 of the Supplementary Material we show that

$$\delta_j\geqslant\frac{q^3}{2^{3n+2}}+\mathcal{O}\left(\frac{q^4}{2^{4n}}\right).$$

The intuition behind this difference is that the queries $i\in I$ satisfy $\lfloor z_i\rfloor_{n-k}=0^{n-k}$, which means that $\lfloor\mathcal{S}_0(x_i)\rfloor_{n-k}\neq 0^{n-k}$. Therefore, these queries do not directly exclude possibilities for $\lfloor y_j\rfloor_{n-k}$ to hit 0^{n-k}, while excluding other options. This slightly increases the probability of $\lfloor y_j\rfloor_{n-k}=0^{n-k}$, leading to the difference. As $\mu_I-\mu_{\varnothing}=\sum_j\delta_j$, this implies that

$$\mu_I-\mu_{\varnothing}\geqslant\frac{q^4}{2^{3n+2}}+\mathcal{O}\left(\frac{q^5}{2^{4n}}\right)=\Omega\left(\frac{q^4}{2^{3n}}\right),$$
$$\frac{1}{(\mu_I-\mu_{\varnothing})^2}=\mathcal{O}\left(\frac{2^{6n}}{q^8}\right).$$

Furthermore, in Lemma 9 of the Supplementary Material we show that the expectation for a single event in both cases is

$$p_{\varnothing,j},p_{I,j}=\frac{q}{2^n}+\mathcal{O}\left(\frac{q^3}{2^{3n}}\right),$$

immediately giving its variance of at most the same value. Furthermore, the variance of a sum of variables is the sum of the variables, with the pairwise correlations added. In our case the variables are negatively correlated as when $\lfloor y_j \rfloor_{n-k} = 0^{n-k}$ happens one more possibility to hit is discarded for future queries, reducing its probability. This means that we can upper bound the variances as

$$\sigma_I^2, \sigma_\varnothing^2 \leqslant \frac{q^2}{2^n} + \mathcal{O}\left(\frac{q^4}{2^{3n}}\right) = \mathcal{O}\left(\frac{q^2}{2^n}\right).$$

Finally, using Lemma 6 of the Supplementary Material we get that the advantage is at least

$$(10) \geqslant 1 - \frac{4(\sigma_I^2 + \sigma_\varnothing^2)}{(\mu_I - \mu_\varnothing)^2} = 1 - \mathcal{O}\left(\frac{q^2}{2^n}\right)\mathcal{O}\left(\frac{2^{6n}}{q^8}\right) = 1 - \mathcal{O}\left(\frac{2^{5n}}{q^6}\right),$$

as desired.

7 Conclusion

The contributions of this work are both negative and positive. On the negative side, we demonstrated that previous best security result on the sum of permutations is flawed and not easily fixed as there is an attack in $2^{5n/6}$ queries. On the positive side, the security claim of the second-best result, guaranteeing $2^{2n/3}/n$ security, can be reattained. The two results, albeit highly technical and non-trivial, admit a gap. We expect that security beyond $2^{2n/3}/n$ may still be possible but that such result will require resorting to a more sophisticated simulator and/or following an entirely different proof approach. Indeed, to be precise, our $2^{2n/3}/n$ security result for the simulator of Definition 5 with $\ell = 2$ took the result of Mennink and Preneel, with $2^{2n/3}/n$ sequential indifferentiability, as a black box and performed a query shuffling approach that was valid as long as the number of queries is at most $2^{2n/3}/n$. Going beyond this security bound thus requires resolving *both* bottlenecks.

Acknowledgements. This work was partly performed while the authors were visiting Dagstuhl Seminar 22141 "Symmetric Cryptography". Aldo Gunsing is supported by the Netherlands Organisation for Scientific Research (NWO) under TOP grant TOP1.18.002 SCALAR. Ritam Bhaumik carried out part of this research while affiliated to Inria Paris, funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement no. 714294 - acronym QUASYModo). Ashwin Jha carried out this work in the framework of the French-German-Center for Cybersecurity, a collaboration of CISPA and LORIA. Bart Mennink is supported by the Netherlands Organisation for Scientific Research (NWO) under grant VI.Vidi.203.099. Yaobin Shen is supported by the European Union through the ERC consolidator grant SWORD (num. 724725).

References

1. Banik, S., Isobe, T., Liu, F., Minematsu, K., Sakamoto, K.: Orthros: A low-latency PRF. IACR Trans. Symmetric Cryptol. 2021(1), 37–77 (2021), https://doi.org/10.46586/tosc.v2021.i1.37-77
2. Bellare, M., Impagliazzo, R.: A tool for obtaining tighter security analyses of pseudorandom function based constructions, with applications to PRP to PRF conversion. Cryptology ePrint Archive, Report 1999/024 (1999), http://eprint.iacr.org/1999/024
3. Bellare, M., Kilian, J., Rogaway, P.: The Security of Cipher Block Chaining. In: Desmedt, Y. (ed.) Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1994, Proceedings. Lecture Notes in Computer Science, vol. 839, pp. 341–358. Springer (1994), https://doi.org/10.1007/3-540-48658-5_32
4. Bellare, M., Krovetz, T., Rogaway, P.: Luby-Rackoff Backwards: Increasing Security by Making Block Ciphers Non-invertible. In: Nyberg, K. (ed.) Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31 - June 4, 1998, Proceeding. Lecture Notes in Computer Science, vol. 1403, pp. 266–280. Springer (1998), https://doi.org/10.1007/BFb0054132
5. Bellare, M., Rogaway, P.: The Security of Triple Encryption and a Framework for Code-Based Game-Playing Proofs. In: Vaudenay, S. (ed.) Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings. Lecture Notes in Computer Science, vol. 4004, pp. 409–426. Springer (2006), https://doi.org/10.1007/11761679_25
6. Bernstein, D.J.: SURF: simple unpredictable random function. https://cr.yp.to/papers.html#surf (Apr 1997)
7. Bhattacharya, S., Nandi, M.: Full Indifferentiable Security of the Xor of Two or More Random Permutations Using the χ^2 Method. In: Nielsen, J.B., Rijmen, V. (eds.) Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part I. Lecture Notes in Computer Science, vol. 10820, pp. 387–412. Springer (2018), https://doi.org/10.1007/978-3-319-78381-9_15
8. Chang, D., Nandi, M.: A Short Proof of the PRP/PRF Switching Lemma. Cryptology ePrint Archive, Report 2008/078 (2008), http://eprint.iacr.org/2008/078
9. Coron, J., Dodis, Y., Malinaud, C., Puniya, P.: Merkle-Damgård Revisited: How to Construct a Hash Function. In: Shoup, V. (ed.) Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3621, pp. 430–448. Springer (2005), https://doi.org/10.1007/11535218_26
10. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Information Security and Cryptography, Springer (2002), http://dx.doi.org/10.1007/978-3-662-04722-4
11. Dai, W., Hoang, V.T., Tessaro, S.: Information-Theoretic Indistinguishability via the Chi-Squared Method. In: Katz, J., Shacham, H. (eds.) Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III. Lecture Notes in Computer Science, vol. 10403, pp. 497–523. Springer (2017), https://doi.org/10.1007/978-3-319-63697-9_17

12. Dodis, Y., Puniya, P.: Getting the Best Out of Existing Hash Functions; or What if We Are Stuck with SHA? In: Bellovin, S.M., Gennaro, R., Keromytis, A.D., Yung, M. (eds.) Applied Cryptography and Network Security, 6th International Conference, ACNS 2008, New York, NY, USA, June 3-6, 2008. Proceedings. Lecture Notes in Computer Science, vol. 5037, pp. 156–173 (2008), https://doi.org/10.1007/978-3-540-68914-0_10
13. Dodis, Y., Ristenpart, T., Shrimpton, T.: Salvaging merkle-damgård for practical applications. In: Joux, A. (ed.) Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5479, pp. 371–388. Springer (2009), https://doi.org/10.1007/978-3-642-01001-9_22
14. Freedman, D.: A remark on the difference between sampling with and without replacement. Journal of the American Statistical Association 72(359), 681–681 (1977), https://dx.doi.org/10.1080/01621459.1977.10480637
15. Gunsing, A.: Block-Cipher-Based Tree Hashing. In: Dodis, Y., Shrimpton, T. (eds.) Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part IV. Lecture Notes in Computer Science, vol. 13510, pp. 205–233. Springer (2022), https://doi.org/10.1007/978-3-031-15985-5_8
16. Hall, C., Wagner, D.A., Kelsey, J., Schneier, B.: Building PRFs from PRPs. In: Krawczyk, H. (ed.) Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings. Lecture Notes in Computer Science, vol. 1462, pp. 370–389. Springer (1998), https://doi.org/10.1007/BFb0055742
17. Impagliazzo, R., Rudich, S.: Limits on the Provable Consequences of One-way Permutations. In: Goldwasser, S. (ed.) Advances in Cryptology - CRYPTO '88, 8th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1988, Proceedings. Lecture Notes in Computer Science, vol. 403, pp. 8–26. Springer (1988), https://doi.org/10.1007/0-387-34799-2_2
18. Lee, J.: Indifferentiability of the Sum of Random Permutations Toward Optimal Security. IEEE Trans. Inf. Theory 63(6), 4050–4054 (2017), https://doi.org/10.1109/TIT.2017.2679757
19. Lucks, S.: The Sum of PRPs Is a Secure PRF. In: Preneel, B. (ed.) Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding. Lecture Notes in Computer Science, vol. 1807, pp. 470–484. Springer (2000), https://doi.org/10.1007/3-540-45539-6_34
20. Mandal, A., Patarin, J., Nachef, V.: Indifferentiability beyond the Birthday Bound for the Xor of Two Public Random Permutations. In: Gong, G., Gupta, K.C. (eds.) Progress in Cryptology - INDOCRYPT 2010 - 11th International Conference on Cryptology in India, Hyderabad, India, December 12-15, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6498, pp. 69–81. Springer (2010), https://doi.org/10.1007/978-3-642-17401-8_6
21. 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 - 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy, March 19-21, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7194, pp. 285–302. Springer (2012), https://doi.org/10.1007/978-3-642-28914-9_16

22. 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, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004, Proceedings. Lecture Notes in Computer Science, vol. 2951, pp. 21–39. Springer (2004), https://doi.org/10.1007/978-3-540-24638-1_2
23. Maurer, U.M., Tessaro, S.: Domain Extension of Public Random Functions: Beyond the Birthday Barrier. In: Menezes, A. (ed.) Advances in Cryptology - CRYPTO 2007, 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007, Proceedings. Lecture Notes in Computer Science, vol. 4622, pp. 187–204. Springer (2007), https://doi.org/10.1007/978-3-540-74143-5_11
24. Mennink, B., Neves, S.: Optimal PRFs from Blockcipher Designs. IACR Trans. Symmetric Cryptol. 2017(3), 228–252 (2017), https://doi.org/10.13154/tosc.v2017.i3.228-252
25. Mennink, B., Preneel, B.: On the XOR of Multiple Random Permutations. In: Malkin, T., Kolesnikov, V., Lewko, A.B., Polychronakis, M. (eds.) Applied Cryptography and Network Security - 13th International Conference, ACNS 2015, New York, NY, USA, June 2-5, 2015, Revised Selected Papers. Lecture Notes in Computer Science, vol. 9092, pp. 619–634. Springer (2015), https://doi.org/10.1007/978-3-319-28166-7_30
26. Patarin, J.: A Proof of Security in O(2n) for the Xor of Two Random Permutations. In: Safavi-Naini, R. (ed.) Information Theoretic Security, Third International Conference, ICITS 2008, Calgary, Canada, August 10-13, 2008, Proceedings. Lecture Notes in Computer Science, vol. 5155, pp. 232–248. Springer (2008), https://doi.org/10.1007/978-3-540-85093-9_22
27. Patarin, J.: Introduction to Mirror Theory: Analysis of Systems of Linear Equalities and Linear Non Equalities for Cryptography. Cryptology ePrint Archive, Report 2010/287 (2010), http://eprint.iacr.org/2010/287
28. Patarin, J.: Security in $O(2^n)$ for the Xor of Two Random Permutations – Proof with the standard H technique–. Cryptology ePrint Archive, Report 2013/368 (2013), http://eprint.iacr.org/2013/368
29. Shrimpton, T., Stam, M.: Building a Collision-Resistant Compression Function from Non-compressing Primitives. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations. Lecture Notes in Computer Science, vol. 5126, pp. 643–654. Springer (2008), https://doi.org/10.1007/978-3-540-70583-3_52
30. Stam, M.: Beyond Uniformity: Better Security/Efficiency Tradeoffs for Compression Functions. In: Wagner, D.A. (ed.) Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings. Lecture Notes in Computer Science, vol. 5157, pp. 397–412. Springer (2008), https://doi.org/10.1007/978-3-540-85174-5_22
31. Yoneyama, K., Miyagawa, S., Ohta, K.: Leaky random oracle. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. 92-A(8), 1795–1807 (2009), https://doi.org/10.1587/transfun.E92.A.1795

Supplementary Material

A Technical Lemmas

Lemma 2. *For any q and any simulator $\mathcal{S}$ we have*

$$\mathbf{Adv}^{\text{indif}}_{F,\mathcal{S}}(q,q) \leqslant \mathbf{Adv}^{\text{seq-indif}}_{F,\mathcal{S}}(q,q) + \mathsf{SD}(q,\mathcal{S}) + \mathsf{FP}(q,\mathcal{S}).$$

Proof. Fix q, $\mathcal{S}$, and $\mathcal{D} := \mathcal{D}^q$. We have

$$\begin{aligned}\mathbf{Adv}^{\text{indif}}_{F,\mathcal{S}}(\mathcal{D}) &:= \left|\mathbb{P}\left[\mathcal{A}^{\mathcal{R},\mathcal{S}} = 1\right] - \mathbb{P}\left[\mathcal{A}^{F,\Pi} = 1\right]\right| \\ &= \left|\mathbb{P}_{\text{id}}\left[\mathcal{T}^{\mathcal{D}\hookrightarrow 1}\right] - \mathbb{P}_{\text{re}}\left[\mathcal{T}^{\mathcal{D}\hookrightarrow 1}\right]\right|. \end{aligned} \tag{11}$$

Claim. We have the bound

$$\mathbf{Adv}^{\text{indif}}_{F,\mathcal{S}}(\mathcal{D}) \leqslant \mathbb{P}_{\text{id}}\left[\mathcal{T}^{\mathcal{D}\geqslant}\right] - \mathbb{P}_{\text{re}}\left[\mathcal{T}^{\mathcal{D}\geqslant}\right]. \tag{12}$$

Proof (of Claim). First consider the case when $\mathbb{P}_{\text{id}}\left[\mathcal{T}^{\mathcal{D}\hookrightarrow 1}\right] \geq \mathbb{P}_{\text{re}}\left[\mathcal{T}^{\mathcal{D}\hookrightarrow 1}\right]$. Then from (11) we have

$$\begin{aligned}\mathbf{Adv}^{\text{indif}}_{F,\mathcal{S}}(\mathcal{D}) &= \mathbb{P}_{\text{id}}\left[\mathcal{T}^{\mathcal{D}\hookrightarrow 1}\right] - \mathbb{P}_{\text{re}}\left[\mathcal{T}^{\mathcal{D}\hookrightarrow 1}\right] \\ &= \sum_{\tau\in\mathcal{T}^{\mathcal{D}\hookrightarrow 1}} \mathbb{P}_{\text{id}}[\tau] - \sum_{\tau\in\mathcal{T}^{\mathcal{D}\hookrightarrow 1}} \mathbb{P}_{\text{re}}[\tau] \\ &= \sum_{\tau\in\mathcal{T}^{\mathcal{D}\hookrightarrow 1}\cap\mathcal{T}^{\mathcal{D}\geqslant}} \left(\mathbb{P}_{\text{id}}[\tau] - \mathbb{P}_{\text{re}}[\tau]\right) \\ &\qquad + \sum_{\tau\in\mathcal{T}^{\mathcal{D}\hookrightarrow 1}\setminus\mathcal{T}^{\mathcal{D}\geqslant}} \left(\mathbb{P}_{\text{id}}[\tau] - \mathbb{P}_{\text{re}}[\tau]\right).\end{aligned}$$

By definition of $\mathcal{T}^{\mathcal{D}\geqslant}$, we have $\mathbb{P}_{\text{id}}[\tau] \leqslant \mathbb{P}_{\text{re}}[\tau]$ for all $\tau \notin \mathcal{T}^{\mathcal{D}\geqslant}$. Thus,

$$\begin{aligned}\mathbf{Adv}^{\text{indif}}_{F,\mathcal{S}}(\mathcal{D}) &\leqslant \sum_{\tau\in\mathcal{T}^{\mathcal{D}\hookrightarrow 1}\cap\mathcal{T}^{\mathcal{D}\geqslant}} \left(\mathbb{P}_{\text{id}}[\tau] - \mathbb{P}_{\text{re}}[\tau]\right) \\ &\leqslant \sum_{\tau\in\mathcal{T}^{\mathcal{D}\geqslant}} \left(\mathbb{P}_{\text{id}}[\tau] - \mathbb{P}_{\text{re}}[\tau]\right) = \mathbb{P}_{\text{id}}\left[\mathcal{T}^{\mathcal{D}\geqslant}\right] - \mathbb{P}_{\text{re}}\left[\mathcal{T}^{\mathcal{D}\geqslant}\right].\end{aligned}$$

Next consider the case when $\mathbb{P}_{\text{id}}\left[\mathcal{T}^{\mathcal{D}\hookrightarrow 1}\right] < \mathbb{P}_{\text{re}}\left[\mathcal{T}^{\mathcal{D}\hookrightarrow 1}\right]$. Then from (11) we have

$$\begin{aligned}\mathbf{Adv}^{\text{indif}}_{F,\mathcal{S}}(\mathcal{D}) &= \mathbb{P}_{\text{re}}\left[\mathcal{T}^{\mathcal{D}\hookrightarrow 1}\right] - \mathbb{P}_{\text{id}}\left[\mathcal{T}^{\mathcal{D}\hookrightarrow 1}\right] \\ &= \left(1 - \mathbb{P}_{\text{re}}\left[\mathcal{T}^{\mathcal{D}\hookrightarrow 0}\right]\right) - \left(1 - \mathbb{P}_{\text{id}}\left[\mathcal{T}^{\mathcal{D}\hookrightarrow 1}\right]\right) \\ &= \mathbb{P}_{\text{id}}\left[\mathcal{T}^{\mathcal{D}\hookrightarrow 0}\right] - \mathbb{P}_{\text{re}}\left[\mathcal{T}^{\mathcal{D}\hookrightarrow 0}\right] \\ &= \sum_{\tau\in\mathcal{T}^{\mathcal{D}\hookrightarrow 0}} \mathbb{P}_{\text{id}}[\tau] - \sum_{\tau\in\mathcal{T}^{\mathcal{D}\hookrightarrow 0}} \mathbb{P}_{\text{re}}[\tau] \\ &= \sum_{\tau\in\mathcal{T}^{\mathcal{D}\hookrightarrow 0}\cap\mathcal{T}^{\mathcal{D}\geqslant}} \left(\mathbb{P}_{\text{id}}[\tau] - \mathbb{P}_{\text{re}}[\tau]\right)\end{aligned}$$

$$
\begin{aligned}
&\quad + \sum_{\tau \in \mathcal{T}^{\mathcal{D}\hookrightarrow 0} \setminus \mathcal{T}^{\mathcal{D}\geqslant}} \left(\mathbb{P}_{\mathrm{id}}\left[\tau\right] - \mathbb{P}_{\mathrm{re}}\left[\tau\right]\right) \\
&\leqslant \sum_{\tau \in \mathcal{T}^{\mathcal{D}\hookrightarrow 0} \cap \mathcal{T}^{\mathcal{D}\geqslant}} \left(\mathbb{P}_{\mathrm{id}}\left[\tau\right] - \mathbb{P}_{\mathrm{re}}\left[\tau\right]\right) \\
&\leqslant \sum_{\tau \in \mathcal{T}^{\mathcal{D}\geqslant}} \left(\mathbb{P}_{\mathrm{id}}\left[\tau\right] - \mathbb{P}_{\mathrm{re}}\left[\tau\right]\right) = \mathbb{P}_{\mathrm{id}}\left[\mathcal{T}^{\mathcal{D}\geqslant}\right] - \mathbb{P}_{\mathrm{re}}\left[\mathcal{T}^{\mathcal{D}\geqslant}\right],
\end{aligned}
$$

thus establishing the claim. □

From (12) we continue

$$
\begin{aligned}
\mathbf{Adv}^{\mathrm{indif}}_{F,\mathcal{S}}(\mathcal{D}) &\leqslant \mathbb{P}_{\mathrm{id}}\left[\mathcal{T}^{\mathcal{D}\geqslant}\right] - \mathbb{P}_{\mathrm{re}}\left[\mathcal{T}^{\mathcal{D}\geqslant}\right] \\
&= \mathbb{P}_{\mathrm{id}}\left[\mathcal{T}^{\mathcal{D}\geqslant}_{\mathrm{good}}\right] - \mathbb{P}_{\mathrm{re}}\left[\mathcal{T}^{\mathcal{D}\geqslant}_{\mathrm{good}}\right] + \mathbb{P}_{\mathrm{id}}\left[\mathcal{T}^{\mathcal{D}\geqslant}_{\mathrm{bad}}\right] - \mathbb{P}_{\mathrm{re}}\left[\mathcal{T}^{\mathcal{D}\geqslant}_{\mathrm{bad}}\right]. \qquad (13)
\end{aligned}
$$

For the last two terms in (13), we have the bound

$$
\mathbb{P}_{\mathrm{id}}\left[\mathcal{T}^{\mathcal{D}\geqslant}_{\mathrm{bad}}\right] - \mathbb{P}_{\mathrm{re}}\left[\mathcal{T}^{\mathcal{D}\geqslant}_{\mathrm{bad}}\right] \leqslant \mathbb{P}_{\mathrm{id}}\left[\mathcal{T}^{\mathcal{D}\geqslant}_{\mathrm{bad}}\right] \leqslant \mathbb{P}_{\mathrm{id}}\left[\mathcal{T}^{\mathcal{D}}_{\mathrm{bad}}\right] \leqslant \mathsf{FP}\left(q, \mathcal{S}\right). \qquad (14)
$$

To bound the first two terms in (13), we first note that for any τ, $\mathbb{P}_{\mathrm{re}}\left[\tau\right] = \mathbb{P}_{\mathrm{re}}\left[\widehat{\tau}\right]$, since the sampling in the real oracle does not depend on the order of queries. Thus we have

$$
\begin{aligned}
&\quad \mathbb{P}_{\mathrm{id}}\left[\mathcal{T}^{\mathcal{D}\geqslant}_{\mathrm{good}}\right] - \mathbb{P}_{\mathrm{re}}\left[\mathcal{T}^{\mathcal{D}\geqslant}_{\mathrm{good}}\right] \\
&= \sum_{\tau \in \mathcal{T}^{\mathcal{D}\geqslant}_{\mathrm{good}}} \left(\mathbb{P}_{\mathrm{id}}\left[\tau\right] - \mathbb{P}_{\mathrm{re}}\left[\tau\right]\right) \\
&= \sum_{\tau \in \mathcal{T}^{\mathcal{D}\geqslant}_{\mathrm{good}}} \left(\mathbb{P}_{\mathrm{id}}\left[\widehat{\tau}\right] - \mathbb{P}_{\mathrm{re}}\left[\widehat{\tau}\right] + \mathbb{P}_{\mathrm{id}}\left[\tau\right] - \mathbb{P}_{\mathrm{id}}\left[\widehat{\tau}\right]\right) \\
&= \sum_{\tau \in \mathcal{T}^{\mathcal{D}\geqslant}_{\mathrm{good}}} \left(\mathbb{P}_{\mathrm{id}}\left[\widehat{\tau}\right] - \mathbb{P}_{\mathrm{re}}\left[\widehat{\tau}\right]\right) + \sum_{\tau \in \mathcal{T}^{\mathcal{D}\geqslant}_{\mathrm{good}}} \left(\mathbb{P}_{\mathrm{id}}\left[\tau\right] - \mathbb{P}_{\mathrm{id}}\left[\widehat{\tau}\right]\right) \\
&\leqslant \sum_{\tau \in \mathcal{T}^{\mathcal{D}\geqslant}_{\mathrm{good}}} \left(\mathbb{P}_{\mathrm{id}}\left[\widehat{\tau}\right] - \mathbb{P}_{\mathrm{re}}\left[\widehat{\tau}\right]\right) + \mathsf{SD}\left(q, \mathcal{S}\right). \qquad (15)
\end{aligned}
$$

We consider an adversary $\widehat{\mathcal{D}}$ which calls $\mathcal{D}_{\mathtt{int}}$ and plays a sequential indifferentiability game as follows (at any point in the game, we call a construction query x *fresh* if it's not yet part of a primitive query-response triple (x, y_0, y_1)):

- When $\mathcal{D}_{\mathtt{int}}$ makes a primitive query, $\widehat{\mathcal{D}}_{\mathtt{int}}$ passes it to the oracle and passes the response back to $\mathcal{D}_{\mathtt{int}}$;
- When $\mathcal{D}_{\mathtt{int}}$ makes a fresh construction query x, $\widehat{\mathcal{D}}_{\mathtt{int}}$ adds it to a queue (initialized as empty), and makes a forward primitive query x to the oracle; it receives the triple (x, y_0, y_1), and returns $(x, y_0 \oplus y_1)$ as the query-response pair to $\mathcal{D}_{\mathtt{int}}$;

– When $\mathcal{D}_{\mathtt{int}}$ makes a construction query x that is not fresh, $\widehat{\mathcal{D}}_{\mathtt{int}}$ simply adds it to the queue of construction queries;
– Once the queries of $\mathcal{D}_{\mathtt{int}}$ have been exhausted, $\widehat{\mathcal{D}}_{\mathtt{int}}$ releases the construction queries from the queue in order;
– Finally, $\widehat{\mathcal{D}}_{\mathtt{dist}}$ examines the transcript from $\widehat{\mathcal{D}}_{\mathtt{int}}$ and outputs a guess bit (which may or may not coincide with the output bit of $\mathcal{D}_{\mathtt{dist}}$.

It is easy to verify that, for a fixed random coin of the oracle, if $\mathcal{D}$ ends up with the transcript τ, $\widehat{\mathcal{D}}$ will end up with the transcript $\widehat{\tau}$. Define

$$\mathcal{T}_*^{\mathcal{D}} := \{\widehat{\tau} \mid \tau \in \mathcal{T}_{\text{good}}^{\mathcal{D}\geqslant}\}.$$

Since the map $\tau \mapsto \widehat{\tau}$ is injective on $\mathcal{T}^{\mathcal{D}}$ (and hence bijective from $\mathcal{T}_{\text{good}}^{\mathcal{D}\geqslant}$ to $\mathcal{T}_*^{\mathcal{D}}$), we have

$$\sum_{\tau \in \mathcal{T}_{\text{good}}^{\mathcal{D}\geqslant}} \left(\mathbb{P}_{\text{id}}\left[\widehat{\tau}\right] - \mathbb{P}_{\text{re}}\left[\widehat{\tau}\right]\right) = \sum_{\tau' \in \mathcal{T}_*^{\mathcal{D}}} \left(\mathbb{P}_{\text{id}}\left[\tau'\right] - \mathbb{P}_{\text{re}}\left[\tau'\right]\right). \tag{16}$$

Now, since $\mathcal{T}_*^{\mathcal{D}} \subseteq \mathcal{T}^{\widehat{\mathcal{D}}}$, we can set $\widehat{\mathcal{D}}_{\mathtt{dist}}$ to output 1 exactly on the set $\mathcal{T}_*^{\mathcal{D}}$. Then we have

$$\begin{aligned}
\sum_{\tau' \in \mathcal{T}_*^{\mathcal{D}}} \left(\mathbb{P}_{\text{id}}\left[\tau'\right] - \mathbb{P}_{\text{re}}\left[\tau'\right]\right) &= \sum_{\tau' \in \mathcal{T}^{\widehat{\mathcal{D}}\hookrightarrow 1}} \left(\mathbb{P}_{\text{id}}\left[\tau'\right] - \mathbb{P}_{\text{re}}\left[\tau'\right]\right) \\
&= \mathbb{P}_{\text{id}}\left[\mathcal{T}^{\widehat{\mathcal{D}}\hookrightarrow 1}\right] - \mathbb{P}_{\text{re}}\left[\mathcal{T}^{\widehat{\mathcal{D}}\hookrightarrow 1}\right] \\
&= \left|\mathbb{P}_{\text{id}}\left[\mathcal{T}^{\widehat{\mathcal{D}}\hookrightarrow 1}\right] - \mathbb{P}_{\text{re}}\left[\mathcal{T}^{\widehat{\mathcal{D}}\hookrightarrow 1}\right]\right| \\
&\leqslant \mathbf{Adv}_{F,\mathcal{S}}^{\text{seq-indif}}\left(\widehat{\mathcal{D}}\right) \\
&\leqslant \mathbf{Adv}_{F,\mathcal{S}}^{\text{seq-indif}}(q,q).
\end{aligned} \tag{17}$$

Combining (15)-(17) gives the bound

$$\mathbb{P}_{\text{id}}\left[\mathcal{T}_{\text{good}}^{\mathcal{D}\geqslant}\right] - \mathbb{P}_{\text{re}}\left[\mathcal{T}_{\text{good}}^{\mathcal{D}\geqslant}\right] \leqslant \mathbf{Adv}_{F,\mathcal{S}}^{\text{seq-indif}}(q,q) + \mathsf{SD}\left(q,\mathcal{S}\right). \tag{18}$$

Using the bounds from (14) and (18) in (13) and taking maximum over $\mathcal{D}$ establishes the lemma. □

Lemma 3. *For the Mennink-Preneel Simulator $\mathcal{S}$ we have*

$$\mathsf{FP}\left(q,\mathcal{S}\right) \leqslant \frac{2q}{2^n} + \frac{13q^3}{2^{2n}}.$$

Proof. Recall that

$$\mathsf{FP}\left(q,\mathcal{S}\right) := \max_{\mathcal{D}} \mathbb{P}_{\text{id}}^{\mathcal{S}}\left[\mathcal{T}_{\text{bad}}^{\mathcal{D}}(q,\mathcal{S})\right].$$

q and $\mathcal{S}$ will be fixed in the following discussion, so we can drop the explicit references to them from our notation. Fix an adversary $\mathcal{D}$. For each $i \in [2q]$, define

$$\mathcal{T}_{\text{bad}_i}^{\mathcal{D}} := \{\tau \in \mathcal{T}_{\text{bad}}^{\mathcal{D}} \mid i \in \mathscr{a}(\tau) \text{ and } [1..i-1] \cap \mathscr{a}(\tau) = \varnothing\},$$

the set of transcripts in $\mathcal{T}^{\mathcal{D}}_{\mathrm{bad}}$ where i-th query is the first query which causes the simulator to be aborted. Then, $\mathcal{T}^{\mathcal{D}}_{\mathrm{bad}} = \sqcup_{i\in[q]}\mathcal{T}^{\mathcal{D}}_{\mathrm{bad}_i}$, whence we have

$$\mathbb{P}_{\mathrm{id}}\left[\mathcal{T}^{\mathcal{D}}_{\mathrm{bad}}\right] = \sum_{\tau\in\mathcal{T}^{\mathcal{D}}_{\mathrm{bad}}} \mathbb{P}_{\mathrm{id}}\left[\tau\right] = \sum_{i\in[q]}\sum_{\tau\in\mathcal{T}^{\mathcal{D}}_{\mathrm{bad}_i}} \mathbb{P}_{\mathrm{id}}\left[\tau\right]. \tag{19}$$

For any $\tau = (\tau_1, \ldots, \tau_{2q})$ and any $i \in [2q]$, let $\mathbb{P}_{\mathrm{id}}\left[\tau_{<i}\right]$ denote the probability of obtaining the partial transcript $(\tau_1, \ldots, \tau_{i-1})$ in the first $i-1$ queries, and let $\mathbb{P}_{\mathrm{id}}\left[\tau_i \mid \tau_{<i}\right]$ denote the conditional probability of getting τ_i at the i-th query having obtained $\tau_1, \ldots, \tau_{i-1}$ in the first $i-1$ queries. ($\mathbb{P}_{\mathrm{id}}\left[\tau_1 \mid \tau_{<1}\right]$ is simply defined as $\mathbb{P}_{\mathrm{id}}\left[\tau_1\right]$.) Finally, let $\mathbb{P}_{\mathrm{id}}\left[\tau_{>i} \mid \tau_{\leqslant i}\right]$ denote the conditional probability of obtaining the partial transcript $(\tau_{i+1}, \ldots, \tau_{2q})$ from the last $2q-n$ queries having obtained $\tau_1, \ldots, \tau_i$ in the first i queries.

Recall that $\mathcal{S}$ can only abort on an inverse simulator query. Let $\mathcal{I}$ denote the indices where $\mathcal{S}$ receives an inverse query. Using this new notation, we can rewrite (19) as

$$\sum_{i\in[2q]}\sum_{\tau\in\mathcal{T}^{\mathcal{D}}_{\mathrm{bad}_i}} \mathbb{P}_{\mathrm{id}}\left[\tau\right] = \sum_{i\in\mathcal{I}}\sum_{\tau\in\mathcal{T}^{\mathcal{D}}_{\mathrm{bad}_i}} \mathbb{P}_{\mathrm{id}}\left[\tau_{<i}\right]\mathbb{P}_{\mathrm{id}}\left[\tau_i \mid \tau_{<i}\right]\mathbb{P}_{\mathrm{id}}\left[\tau_{>i} \mid \tau_{\leqslant i}\right] \tag{20}$$

We'll concentrate on the conditional probability $\mathbb{P}_{\mathrm{id}}\left[\tau_i \mid \tau_{<i}\right]$. Without loss of generality assume that the adversary made an inverse query y_0 to the simulator interface $\mathcal{S}_0^-$, and the simulator returned $(y_0, \perp)$. This is possible if and only if the simulator fails to sample a valid preimage x (at line 2. (a) ii.) on two consecutive attempts. Let x^0 and x^1 denote the two sampled preimages. For $0 \leqslant j \leqslant 2$, let $\mathtt{E}_j$ denote the event $|\{x^0, x^1\} \cap \{x \mid (x, z) \in \tau_{<i}\}| = j$. Then

$$\begin{aligned}\mathbb{P}_{\mathrm{id}}\left[\tau_i \mid \tau_{<i}\right] &= \mathbb{P}_{\mathrm{id}}\left[\tau_i, \mathtt{E}_0 \mid \tau_{<i}\right] + \mathbb{P}_{\mathrm{id}}\left[\tau_i, \mathtt{E}_1 \mid \tau_{<i}\right] + \mathbb{P}_{\mathrm{id}}\left[\tau_i, \mathtt{E}_2 \mid \tau_{<i}\right] \\ &\leqslant \mathbb{P}_{\mathrm{id}}\left[\tau_i, \mathtt{E}_0 \mid \tau_{<i}\right] + \mathbb{P}_{\mathrm{id}}\left[\tau_i, \mathtt{E}_1 \mid \tau_{<i}\right] + \mathbb{P}_{\mathrm{id}}\left[\mathtt{E}_2 \mid \tau_{<i}\right]. \end{aligned} \tag{21}$$

We will handle one by one the three terms on the right hand side of (21). Let D and R_1 respectively denote the shared domain of P_0 and P_1 and the range of P_1 before the i-th query. We will use the bounds $|\mathsf{D}| \leqslant q \leqslant 2^{n-1}$ (and thus $2^n - |\mathsf{D}| \geq 2^{n-1}$), and $|\mathsf{R}_1| \leqslant q$.

Bounding $\mathbb{P}_{\mathrm{id}}\left[\tau_i, \mathtt{E}_0 \mid \tau_{<i}\right]$. Let $\mathtt{D}$ denote the event that $x^0 = x^1$. Then, we have

$$\begin{aligned}\mathbb{P}_{\mathrm{id}}\left[\tau_i, \mathtt{E}_0 \mid \tau_{<i}\right] &= \mathbb{P}_{\mathrm{id}}\left[\tau_i, \mathtt{E}_0, \mathtt{D} \mid \tau_{<i}\right] + \mathbb{P}_{\mathrm{id}}\left[\tau_i, \mathtt{E}_0, \neg\mathtt{D} \mid \tau_{<i}\right] \\ &\leqslant \mathbb{P}_{\mathrm{id}}\left[\mathtt{D} \mid \tau_{<i}\right] + \mathbb{P}_{\mathrm{id}}\left[\tau_i \mid \tau_{<i}, \mathtt{E}_0, \neg\mathtt{D}\right] \\ &\leqslant \frac{1}{2^n - |\mathsf{D}|} + \mathbb{P}_{\mathrm{id}}\left[\tau_i \mid \tau_{<i}, \mathtt{E}_0, \neg\mathtt{D}\right] \\ &\leqslant \frac{2}{2^n} + \mathbb{P}_{\mathrm{id}}\left[\tau_i \mid \tau_{<i}, \mathtt{E}_0, \neg\mathtt{D}\right] \end{aligned} \tag{22}$$

where the second inequality follows from the fact that x^0 and x^1 are sampled uniformly at random from a set of size $2^n - |\mathsf{D}|$. Now, assume $\neg\mathtt{D}$ and $\mathtt{E}_0$ holds,

i.e., $x^0 \neq x^1$ and neither has been queried to the construction before. Then we have

$$\begin{aligned}\mathbb{P}_{\mathrm{id}}\left[\tau_i \mid \tau_{<i}, \mathtt{E}_0, \neg\mathtt{D}\right] &\leqslant \mathbb{P}_{\mathrm{id}}\left[\mathcal{R}(x_0), \mathcal{R}(x_1) \in \mathsf{R}_1 \oplus y_0 \mid \tau_{<i}, \mathtt{E}_0, \neg\mathtt{D}\right] \\ &\leqslant \frac{|\mathsf{R}_1 \oplus y_0|}{2^n} \cdot \frac{|\mathsf{R}_1 \oplus y_0|}{2^n} \\ &= \frac{|\mathsf{R}_1|}{2^n} \cdot \frac{|\mathsf{R}_1|}{2^n} \leqslant \frac{q^2}{2^{2n}}, \end{aligned} \tag{23}$$

Using the bound from (23) in (22) yields

$$\mathbb{P}_{\mathrm{id}}\left[\tau_i, \mathtt{E}_0 \mid \tau_{<i}\right] \leqslant \frac{2}{2^n} + \frac{q^2}{2^{2n}}. \tag{24}$$

Bounding $\mathbb{P}_{\mathrm{id}}\left[\tau_i, \mathtt{E}_1 \mid \tau_{<i}\right]$. For $a \in \{0,1\}$, let $\mathtt{F}^a$ denote the event that there exists an $i' < i$ such that $\tau_{i'} = (x^a, z)$. We can decompose the event as

$$\begin{aligned}\mathbb{P}_{\mathrm{id}}\left[\tau_i, \mathtt{E}_1 \mid \tau_{<i}\right] &= \sum_{a=0}^{1} \mathbb{P}_{\mathrm{id}}\left[\tau_i, \mathtt{E}_1, \mathtt{F}^a \mid \tau_{<i}\right] \\ &\leqslant \sum_{a=0}^{1} \mathbb{P}_{\mathrm{id}}\left[\mathtt{E}_1, \mathtt{F}^a \mid \tau_{<i}\right] \cdot \mathbb{P}_{\mathrm{id}}\left[\tau_i \mid \tau_{<i}, \mathtt{E}_1, \mathtt{F}^a\right]. \end{aligned} \tag{25}$$

For $a = 0$, x^0 is sampled uniformly at random from a set of size $2^n - |\mathsf{D}|$, and there are at most q choices for $i' < i$ such that $\tau_{i'} = (x^0, z)$ for some z. Thus,

$$\mathbb{P}_{\mathrm{id}}\left[\mathtt{E}_1, \mathtt{F}^0 \mid \tau_{<i}\right] \leqslant \frac{q}{2^n - |\mathsf{D}|}.$$

Further, given $\mathtt{E}_1$ and $\mathtt{F}^0$, arguing as in case 1 above, we have

$$\mathbb{P}_{\mathrm{id}}\left[\tau_i \mid \tau_{<i}, \mathtt{E}_1, \mathtt{F}^0\right] \leqslant \frac{|\mathsf{R}_1 \oplus y_0|}{2^n} = \frac{|\mathsf{R}_1|}{2^n}.$$

For $a = 1$ by symmetry we obtain the exact same bounds. Substituting these bounds in (25), we have

$$\mathbb{P}_{\mathrm{id}}\left[\tau_i, \mathtt{E}_1 \mid \tau_{<i}\right] \leqslant 2 \cdot \frac{q}{2^n - |\mathsf{D}|} \cdot \frac{|\mathsf{R}_1|}{2^n} \leqslant \frac{4q^2}{2^{2n}}. \tag{26}$$

Bounding $\mathbb{P}_{\mathrm{id}}\left[\mathtt{E}_2 \mid \tau_{<i}\right]$. Since x^0 and x^1 are chosen uniformly at random from a set of size $2^n - |\mathsf{D}|$, and there are at most q construction queries, we have

$$\mathbb{P}_{\mathrm{id}}\left[\mathtt{E}_2 \mid \tau_{<i}\right] \leqslant \left(\frac{q_c}{2^n - |\mathsf{D}|}\right)^2 \leqslant \frac{4q^2}{2^{2n}}. \tag{27}$$

Using the bounds in (21), (24), (26) and (27) in (20) and applying the summation, we have

$$\sum_{i \in [2q]} \sum_{\tau \in \mathcal{T}^{\mathcal{D}}_{\mathrm{bad}_i}} \mathbb{P}_{\mathrm{id}}\left[\tau\right] \leqslant \frac{2|\mathcal{I}|}{2^n} + \frac{13q^2|\mathcal{I}|}{2^{2n}}. \tag{28}$$

Substituting the bound from (28) in (19) and observing that $|\mathcal{I}| \leqslant q$ completes the proof. □

Lemma 4. *Using the notation in Section 4 we have*

$$\mathbb{P}_{\mathrm{id}}\left[\tau_i \mid \tau_{\mathit{head}}\right] = \frac{\left|\mathsf{S}_{\mathsf{D_C}\setminus\mathsf{D}\to\mathsf{R}_1\oplus y_0}\right| + 2^n - q|\mathsf{D}|/2^n}{(2^n - |\mathsf{D}|)^2}.$$

Proof. Suppose at the first attempt (call it A_0) the simulator samples x^0. Then we have

$$\begin{aligned}
&\mathbb{P}_{\mathrm{id}}\left[A_0 \text{ fails} \mid \tau_{\mathtt{head}}\right] \\
&= \mathbb{P}_{\mathrm{id}}\left[A_0 \text{ fails}, x^0 \in \mathsf{D_C} \mid \tau_{\mathtt{head}}\right] + \mathbb{P}_{\mathrm{id}}\left[A_0 \text{ fails}, x^0 \notin \mathsf{D_C} \mid \tau_{\mathtt{head}}\right] \\
&= \mathbb{P}_{\mathrm{id}}\left[A_0 \text{ fails} \mid x^0 \in \mathsf{D_C}, \tau_{\mathtt{head}}\right] \cdot \mathbb{P}_{\mathrm{id}}\left[x^0 \in \mathsf{D_C} \mid \tau_{\mathtt{head}}\right] \\
&\qquad\qquad + \mathbb{P}_{\mathrm{id}}\left[A_0 \text{ fails} \mid x^0 \notin \mathsf{D_C}, \tau_{\mathtt{head}}\right] \cdot \mathbb{P}_{\mathrm{id}}\left[x^0 \notin \mathsf{D_C} \mid \tau_{\mathtt{head}}\right] \\
&= \mathbb{P}_{\mathrm{id}}\left[x^0 \in \mathsf{S}_{\mathsf{D_C}\setminus\mathsf{D}\to\mathsf{R}_1\oplus y_0} \mid x^0 \in \mathsf{D_C}, \tau_{\mathtt{head}}\right] \cdot \frac{|\mathsf{D_C}| - |\mathsf{D}|}{2^n - |\mathsf{D}|} \\
&\qquad\qquad + \mathbb{P}_{\mathrm{id}}\left[\$\big(x^0\big) \oplus y_0 \in \mathsf{R}_1 \mid x^0 \notin \mathsf{D_C}, \tau_{\mathtt{head}}\right] \cdot \frac{2^n - |\mathsf{D_C}|}{2^n - |\mathsf{D}|} \\
&= \frac{\left|\mathsf{S}_{\mathsf{D_C}\setminus\mathsf{D}\to\mathsf{R}_1\oplus y_0}\right|}{|\mathsf{D_C}| - |\mathsf{D}|} \cdot \frac{|\mathsf{D_C}| - |\mathsf{D}|}{2^n - |\mathsf{D}|} + \frac{|\mathsf{R}_1|}{2^n} \cdot \frac{2^n - |\mathsf{D_C}|}{2^n - |\mathsf{D}|} \\
&= \frac{\left|\mathsf{S}_{\mathsf{D_C}\setminus\mathsf{D}\to\mathsf{R}_1\oplus y_0}\right|}{2^n - |\mathsf{D}|} + \frac{|\mathsf{D}|}{2^n} \cdot \frac{2^n - |\mathsf{D_C}|}{2^n - |\mathsf{D}|} \\
&= \frac{1}{2^n - |\mathsf{D}|} \cdot \left(\left|\mathsf{S}_{\mathsf{D_C}\setminus\mathsf{D}\to\mathsf{R}_1\oplus y_0}\right| + |\mathsf{D}| - \frac{|\mathsf{D}| \cdot |\mathsf{D_C}|}{2^n}\right). \qquad (29)
\end{aligned}$$

Suppose in the second attempt the simulator samples x^1. With the knowledge that if A_0 fails, the second attempt must succeed (since $\tau \in \mathcal{T}_{\mathrm{good}}^{\mathcal{D}\geqslant}$), we can now calculate

$$\begin{aligned}
\mathbb{P}_{\mathrm{id}}\left[\tau_i \mid \tau_{\mathtt{head}}\right] &= \mathbb{P}_{\mathrm{id}}\left[\tau_i, A_0 \text{ succeeds} \mid \tau_{\mathtt{head}}\right] + \mathbb{P}_{\mathrm{id}}\left[\tau_i, A_0 \text{ fails} \mid \tau_{\mathtt{head}}\right] \\
&= \mathbb{P}_{\mathrm{id}}\left[x^0 = x \mid \tau_{\mathtt{head}}\right] \\
&\qquad + \mathbb{P}_{\mathrm{id}}\left[\tau_i \mid A_0 \text{ fails}, \tau_{\mathtt{head}}\right] \cdot \mathbb{P}_{\mathrm{id}}\left[A_0 \text{ fails} \mid \tau_{\mathtt{head}}\right] \\
&= \frac{1}{2^n - |\mathsf{D}|} + \mathbb{P}_{\mathrm{id}}\left[x^1 = x \mid A_0 \text{ fails}, \tau_{\mathtt{head}}\right] \cdot \mathbb{P}_{\mathrm{id}}\left[A_0 \text{ fails} \mid \tau_{\mathtt{head}}\right] \\
&= \frac{1}{2^n - |\mathsf{D}|} + \frac{1}{2^n - |\mathsf{D}|} \cdot \mathbb{P}_{\mathrm{id}}\left[A_0 \text{ fails} \mid \tau_{\mathtt{head}}\right] \\
&= \frac{1}{2^n - |\mathsf{D}|} \cdot \big(1 + \mathbb{P}_{\mathrm{id}}\left[A_0 \text{ fails} \mid \tau_{\mathtt{head}}\right]\big). \qquad (30)
\end{aligned}$$

From (29) we have

$$1 + \mathbb{P}_{\mathrm{id}}\left[A_0 \text{ fails}\right]$$

$$= 1 + \frac{1}{2^n - |\mathsf{D}|} \cdot \left(\left| \mathsf{S}_{\mathsf{D_C} \setminus \mathsf{D} \to \mathsf{R}_1 \oplus y_0} \right| + |\mathsf{D}| - \frac{|\mathsf{D}| \cdot |\mathsf{D_C}|}{2^n} \right)$$
$$= \frac{1}{2^n - |\mathsf{D}|} \cdot \left(\left| \mathsf{S}_{\mathsf{D_C} \setminus \mathsf{D} \to \mathsf{R}_1 \oplus y_0} \right| + 2^n - \frac{q|\mathsf{D}|}{2^n} \right), \tag{31}$$

since $|\mathsf{D_C}| = q$. Substituting (31) in (30) gives

$$\mathbb{P}_{\mathrm{id}} \left[\tau_i \mid \tau_{\mathsf{head}} \right] = \frac{\left| \mathsf{S}_{\mathsf{D_C} \setminus \mathsf{D} \to \mathsf{R}_1 \oplus y_0} \right| + 2^n - q|\mathsf{D}|/2^n}{(2^n - |\mathsf{D}|)^2}.$$

Remark 5. We do not compute any probabilities on getting y_1 correct, as y_1 is always computed correctly once x and y_0 are fixed; for transcripts with wrong y_1 values the probability trivially becomes 0.

Lemma 5. *Using the notation in Section 4, we achieve the following upper bound on* $\rho_\tau^{(i,j)}$*:*

$$\rho_\tau^{(i,j)} \leqslant \begin{cases} 1 + \dfrac{5}{2^{2n}} & \text{if } (\tau_{i^*-j}, \tau_{i^*}) \text{ is erratic,} \\ 1 + \dfrac{5q}{2^{2n}} & \text{otherwise.} \end{cases}$$

Proof. We can consider several cases based on the swapped pair $(\tau_{i^*-j}, \tau_{i^*})$. We note here that since we are implicitly conditioning on the outputs of all the construction queries, whenever τ_{i^*-j} is a construction query, $\rho_\tau^{(i,j)} = 1$, since the order of the two queries won't affect the joint probability of τ_{i^*-j} and τ_{i^*}. Thus it is sufficient to look at the cases where τ_{i^*-j} is a primitive query. This leaves five different cases for $(\tau_{i^*-j}, \tau_{i^*})$:

- both are forward queries (case 1);
- only τ_{i^*-j} is a forward query (case 2);
- only τ_{i^*} is a forward query (case 3);
- both are backward queries to the same primitive (case 4);
- both are backward queries, but to different primitives (case 5).

We examine the cases one by one. Let R_0 and R_1 be the ranges of the partial permutations P_0 and P_1 respectively just before the $(i^* - j)$-th query, i.e., up to the point when $\tau_{\mathsf{head}}^{(i,j)}$ has been obtained, and let D be their shared domain at that point.

Case 1: $\boldsymbol{\tau_{i^*-j} = (x, y_0, y_1)^+}$, $\boldsymbol{\tau_{i^*} = (x', y_0', y_1')^+}$

We obtain from multiple applications of (7) the equations

$$\mathbb{P}_{\mathrm{id}} \left[\tau_{i^*-j} \mid \tau_{\mathsf{head}}^{(i,j)} \right] \cdot \mathbb{P}_{\mathrm{id}} \left[\tau_{i^*} \mid \tau_{i^*-j}, \tau_{\mathsf{head}}^{(i,j)} \right]$$
$$= \frac{1}{2^n - |\mathsf{R}_0 \cup (\mathsf{R}_1 \oplus z)|} \cdot \frac{1}{2^n - |\mathsf{R}_0 \cup (\mathsf{R}_1 \oplus z') \cup \{y_0, y_1 \oplus z'\}|},$$

$$\mathbb{P}_{\mathrm{id}}\left[\tau_{i^*} \mid \tau_{\mathsf{head}}^{(i,j)}\right] \cdot \mathbb{P}_{\mathrm{id}}\left[\tau_{i^*-j} \mid \tau_{i^*}, \tau_{\mathsf{head}}^{(i,j)}\right]$$
$$= \frac{1}{2^n - |\mathsf{R}_0 \cup (\mathsf{R}_1 \oplus z')|} \cdot \frac{1}{2^n - |\mathsf{R}_0 \cup (\mathsf{R}_1 \oplus z) \cup \{y_0', y_1' \oplus z\}|}. \tag{32}$$

Define $\lambda := |\mathsf{R}_0 \cup (\mathsf{R}_1 \oplus z)|$, $\lambda' := |\mathsf{R}_0 \cup (\mathsf{R}_1 \oplus z')|$, $\delta := 2 - |\mathsf{R}_0 \cap \{y_1' \oplus z\}| - |\mathsf{R}_1 \cap \{y_0' \oplus z\}|$, and $\delta' := 2 - |\mathsf{R}_0 \cap \{y_1 \oplus z'\}| - |\mathsf{R}_1 \cap \{y_0 \oplus z'\}|$. It is easy to check that $0 \leqslant \delta, \delta' \leqslant 2$. Then we have

$$\begin{aligned}
&|\mathsf{R}_0 \cup (\mathsf{R}_1 \oplus z') \cup \{y_0, y_1 \oplus z'\}| \\
&= |\mathsf{R}_0 \cup (\mathsf{R}_1 \oplus z')| + |\{y_0, y_1 \oplus z'\}| - \left|[\mathsf{R}_0 \cup (\mathsf{R}_1 \oplus z')] \cap \{y_0, y_1 \oplus z'\}\right| \\
&= \lambda' + 2 - (|\mathsf{R}_0 \cap \{y_1 \oplus z'\}| + |(\mathsf{R}_1 \oplus z') \cap \{y_0\}|) = \lambda' + \delta'.
\end{aligned}$$

Similarly we can show that

$$|\mathsf{R}_0 \cup (\mathsf{R}_1 \oplus z) \cup \{y_0', y_1' \oplus z\}| = \lambda + \delta.$$

Thus (32) becomes

$$\begin{aligned}
\mathbb{P}_{\mathrm{id}}\left[\tau_{i^*-j} \mid \tau_{\mathsf{head}}^{(i,j)}\right] \cdot \mathbb{P}_{\mathrm{id}}\left[\tau_{i^*} \mid \tau_{i^*-j}, \tau_{\mathsf{head}}^{(i,j)}\right] &= \frac{1}{2^n - \lambda} \cdot \frac{1}{2^n - \lambda' - \delta'}, \\
\mathbb{P}_{\mathrm{id}}\left[\tau_{i^*} \mid \tau_{\mathsf{head}}^{(i,j)}\right] \cdot \mathbb{P}_{\mathrm{id}}\left[\tau_{i^*-j} \mid \tau_{i^*}, \tau_{\mathsf{head}}^{(i,j)}\right] &= \frac{1}{2^n - \lambda'} \cdot \frac{1}{2^n - \lambda - \delta}.
\end{aligned} \tag{33}$$

Substituting (33) in (9) yields

$$\begin{aligned}
\rho_\tau^{(i,j)} &= \frac{\mathbb{P}_{\mathrm{id}}\left[\tau_{i^*-j} \mid \tau_{\mathsf{head}}^{(i,j)}\right] \cdot \mathbb{P}_{\mathrm{id}}\left[\tau_{i^*} \mid \tau_{i^*-j}, \tau_{\mathsf{head}}^{(i,j)}\right]}{\mathbb{P}_{\mathrm{id}}\left[\tau_{i^*} \mid \tau_{\mathsf{head}}^{(i,j)}\right] \cdot \mathbb{P}_{\mathrm{id}}\left[\tau_{i^*-j} \mid \tau_{i^*}, \tau_{\mathsf{head}}^{(i,j)}\right]} \\
&= \frac{(2^n - \lambda') \cdot (2^n - \lambda - \delta)}{(2^n - \lambda) \cdot (2^n - \lambda' - \delta')} \\
&= \frac{(2^n - \lambda') \cdot (2^n - \lambda) - (2^n - \lambda') \cdot \delta}{(2^n - \lambda) \cdot (2^n - \lambda' - \delta')} \\
&= \frac{(2^n - \lambda' - \delta') \cdot (2^n - \lambda) + \delta' \cdot (2^n - \lambda) - (2^n - \lambda') \cdot \delta}{(2^n - \lambda) \cdot (2^n - \lambda' - \delta')} \\
&= 1 + \frac{2^n \cdot (\delta' - \delta) - \delta'\lambda + \delta\lambda'}{(2^n - \lambda) \cdot (2^n - \lambda' - \delta')}.
\end{aligned} \tag{34}$$

We can simplify (34) using the bounds $2^n - \lambda \geq 2^{n-1}$, $2^n - \lambda' - \delta' \geq 2^{n-1}$, $\lambda' \leqslant 2q$, $\lambda \geq q$. When $\delta' = \delta$, (34) gives

$$\rho_\tau^{(i,j)} \leqslant 1 + \frac{4q}{2^{2n}}.$$

When $\delta' > \delta$, we use $\delta' - \delta \leqslant 2$ to get

$$\rho_\tau^{(i,j)} \leqslant 1 + \frac{4}{2^n}.$$

We note here that $\delta' > \delta \implies \delta < 2$, and this needs one of the two events $y_1' \oplus z \in \mathsf{R}_0$ and $y_0' \oplus z \in \mathsf{R}_1$ to be true, which can only happen when $(\tau_{i^*-j}, \tau_{i^*})$ is an erratic pair.

Case 2: $\tau_{i^*-j} = (x, y_0, y_1)^+$, $\tau_{i^*} = (x', y'_0, y'_1)^-_b$

Without loss of generality we can assume that $b = 0$. From (7) we have

$$\mathbb{P}_{\mathrm{id}}\left[\tau_{i^*-j} \mid \tau^{(i,j)}_{\mathsf{head}}\right] = \frac{1}{2^n - |\mathsf{R}_0 \cup (\mathsf{R}_1 \oplus z)|}, \tag{35}$$

where $z = \mathsf{C}(x)$. Then from (8) we have

$$\mathbb{P}_{\mathrm{id}}\left[\tau_{i^*} \mid \tau_{i^*-j}, \tau^{(i,j)}_{\mathsf{head}}\right] = \frac{\left|\mathsf{S}_{\mathsf{D_C}\setminus(\mathsf{D}\cup\{x\})\to(\mathsf{R}_1\cup\{y_1\})\oplus y'_0}\right| + 2^n - q|\mathsf{D}\cup\{x\}|/2^n}{(2^n - |\mathsf{D}\cup\{x\}|)^2}. \tag{36}$$

Using the same equations for the swapped order we have

$$\mathbb{P}_{\mathrm{id}}\left[\tau_{i^*} \mid \tau^{(i,j)}_{\mathsf{head}}\right] = \frac{\left|\mathsf{S}_{\mathsf{D_C}\setminus\mathsf{D}\to\mathsf{R}_1\oplus y'_0}\right| + 2^n - q|\mathsf{D}|/2^n}{(2^n - |\mathsf{D}|)^2}, \tag{37}$$

$$\mathbb{P}_{\mathrm{id}}\left[\tau_{i^*-j} \mid \tau_{i^*}, \tau^{(i,j)}_{\mathsf{head}}\right] = \frac{1}{2^n - |\mathsf{R}_0 \cup (\mathsf{R}_1 \oplus z) \cup \{y'_0, y'_1 \oplus z\}|}. \tag{38}$$

With $\mathsf{S} := \mathsf{S}_{\mathsf{D_C}\setminus\mathsf{D}\to\mathsf{R}_1\oplus y'_0}$, $\mathsf{S}' := \mathsf{S}_{\mathsf{D_C}\setminus(\mathsf{D}\cup\{x\})\to(\mathsf{R}_1\cup\{y_1\})\oplus y'_0}$, and λ and δ as in case 1, we have from (35)-(38)

$$\begin{aligned}
\rho^{(i,j)}_\tau &= \frac{\mathbb{P}_{\mathrm{id}}\left[\tau_{i^*-j} \mid \tau^{(i,j)}_{\mathsf{head}}\right] \cdot \mathbb{P}_{\mathrm{id}}\left[\tau_{i^*} \mid \tau_{i^*-j}, \tau^{(i,j)}_{\mathsf{head}}\right]}{\mathbb{P}_{\mathrm{id}}\left[\tau_{i^*} \mid \tau^{(i,j)}_{\mathsf{head}}\right] \cdot \mathbb{P}_{\mathrm{id}}\left[\tau_{i^*-j} \mid \tau_{i^*}, \tau^{(i,j)}_{\mathsf{head}}\right]} \\
&= \frac{2^n - \lambda - \delta}{2^n - \lambda} \cdot \frac{|\mathsf{S}'| + 2^n - q(|\mathsf{D}|+1)/2^n}{|\mathsf{S}| + 2^n - q|\mathsf{D}|/2^n} \cdot \left(\frac{2^n - |\mathsf{D}|}{2^n - |\mathsf{D}| - 1}\right)^2 \\
&= \left(1 - \frac{\delta}{2^n - \lambda}\right)\left(1 + \frac{|\mathsf{S}'| - |\mathsf{S}| - q/2^n}{|\mathsf{S}| + 2^n - q|\mathsf{D}|/2^n}\right)\left(1 + \frac{1}{2^n - |\mathsf{D}| - 1}\right)^2.
\end{aligned} \tag{39}$$

We observe that

$$\mathsf{S}' \setminus \mathsf{S} \subseteq \{x'' \in \mathsf{D_C} \mid \mathsf{C}(x'') = y'_0 \oplus y_1\}. \tag{40}$$

Since τ is well-behaved, the right-hand side of (40) has at most two elements. Thus, $|\mathsf{S}'| - |\mathsf{S}| \leqslant |\mathsf{S}' \setminus \mathsf{S}| \leqslant 2$. Moreover, in order for $\mathsf{S}' \setminus \mathsf{S}$ to be non-empty, $y'_0 \oplus y_1$ needs to be in $\mathsf{R_C}$, which would make $(\tau_{i^*-j}, \tau_{i^*})$ an erratic pair. When $(\tau_{i^*-j}, \tau_{i^*})$ is not erratic, both $|\mathsf{S}' \setminus \mathsf{S}| = 0$ and $\delta = 2$, so using $\lambda \geq |\mathsf{D}|$ we have

$$\begin{aligned}
\rho^{(i,j)}_\tau &\leqslant \left(1 - \frac{2}{2^n - \lambda}\right)\left(1 + \frac{1}{2^n - \lambda - 1}\right)^2 \\
&= \left(1 - \frac{2}{2^n - \lambda}\right)\left(1 + \frac{2}{2^n - \lambda - 1} + \frac{1}{(2^n - \lambda - 1)^2}\right) \\
&= 1 + \frac{2}{2^n - \lambda - 1} + \frac{1}{(2^n - \lambda - 1)^2} - \frac{2}{2^n - \lambda} \\
&\qquad - \frac{4}{(2^n - \lambda)(2^n - \lambda - 1)} - \frac{2}{(2^n - \lambda)(2^n - \lambda - 1)^2}
\end{aligned}$$

$$\leqslant 1 + \frac{1}{(2^n - \lambda - 1)^2} - \frac{2}{(2^n - \lambda)(2^n - \lambda - 1)} \leqslant 1.$$

When either $|\mathsf{S}' \setminus \mathsf{S}| > 0$ or $\delta < 2$ (both of which require $(\tau_{i^*-j}, \tau_{i^*})$ to be erratic) we can use the bound

$$\begin{aligned}\rho_\tau^{(i,j)} &\leqslant \left(1 + \frac{2}{2^n - q^2/2^n}\right)\left(1 + \frac{1}{2^n - q}\right)^2 \\ &\leqslant \left(1 + \frac{2}{2^n - q}\right)\left(1 + \frac{2}{2^n - q} + \frac{1}{(2^n - q)^2}\right) \\ &= 1 + \frac{4}{2^n - q} + \frac{5}{(2^n - q)^2} + \frac{2}{(2^n - q)^3} \leqslant 1 + \frac{5}{2^n},\end{aligned}$$

where for the last inequality we use the assumption $q \leqslant 2^n/9$.

Case 3: $\boldsymbol{\tau_{i^*-j} = (x, y_0, y_1)_b^-}$, $\boldsymbol{\tau_{i^*} = (x', y_0', y_1')^+}$

Again without loss of generality we assume $b = 0$, which makes the $\rho_\tau^{(i,j)}$ in this case the reciprocal of that in Case 2. Writing $\mathsf{S}^\dagger := \mathsf{S}_{\mathsf{D_C} \setminus \mathsf{D} \to \mathsf{R}_1 \oplus y_0}$ and $\mathsf{S}'^\dagger := \mathsf{S}_{\mathsf{D_C} \setminus (\mathsf{D} \cup \{x'\}) \to (\mathsf{R}_1 \cup \{y_1'\}) \oplus y_0}$, and with λ' and δ' as in Case 1, we have

$$\begin{aligned}\rho_\tau^{(i,j)} &= \frac{2^n - \lambda'}{2^n - \lambda' - \delta'} \cdot \frac{|\mathsf{S}^\dagger| + 2^n - q|\mathsf{D}|/2^n}{|\mathsf{S}'^\dagger| + 2^n - q(|\mathsf{D}| + 1)/2^n} \cdot \left(\frac{2^n - |\mathsf{D}| - 1}{2^n - |\mathsf{D}|}\right)^2 \\ &= \left(1 + \frac{\delta'}{2^n - \lambda' - \delta'}\right)\left(1 + \frac{|\mathsf{S}^\dagger| - |\mathsf{S}'^\dagger| + q/2^n}{|\mathsf{S}'^\dagger| + 2^n - q(|\mathsf{D}| + 1)/2^n}\right)\left(1 - \frac{1}{2^n - |\mathsf{D}|}\right)^2. \qquad (41)\end{aligned}$$

The only candidate for $\mathsf{S}^\dagger \setminus \mathsf{S}'^\dagger$ is x', so $|\mathsf{S}^\dagger| - |\mathsf{S}'^\dagger| \leqslant |\mathsf{S}^\dagger \setminus \mathsf{S}'^\dagger| \leqslant 1$. When $|\mathsf{S}^\dagger \setminus \mathsf{S}'^\dagger| = 0$, we use $\delta' \leqslant 2$ and $\lambda' + \delta' \leqslant 2|\mathsf{D}|$ to get

$$\begin{aligned}\rho_\tau^{(i,j)} &\leqslant \left(1 + \frac{2}{2^n - 2|\mathsf{D}|}\right)\left(1 + \frac{q/2^n}{|\mathsf{S}'^\dagger| + 2^n - q(|\mathsf{D}| + 1)/2^n}\right)\left(1 - \frac{1}{2^n - |\mathsf{D}|}\right)^2 \\ &\leqslant \left(1 + \frac{2}{2^n - 2|\mathsf{D}|}\right)\left(1 + \frac{2q}{2^{2n}}\right)\left(1 - \frac{2}{2^n - |\mathsf{D}|} + \frac{1}{(2^n - |\mathsf{D}|)^2}\right) \\ &\leqslant \left(1 + \frac{2|\mathsf{D}|}{(2^n - 2|\mathsf{D}|)(2^n - |\mathsf{D}|)} + \frac{2}{2^{2n}}\right)\left(1 + \frac{2q}{2^{2n}}\right) \\ &\leqslant \left(1 + \frac{2q}{(2^n - 2q)(2^n - q)} + \frac{2}{2^{2n}}\right)\left(1 + \frac{2q}{2^{2n}}\right) \leqslant 1 + \frac{5q}{2^{2n}}.\end{aligned}$$

When $\mathsf{S}^\dagger \setminus \mathsf{S}'^\dagger = \{x'\}$ (which can only happen when $(\tau_{i^*-j}, \tau_{i^*})$ is an erratic pair), we use the bound

$$\rho_\tau^{(i,j)} \leqslant \left(1 + \frac{2}{2^n - 2q}\right)\left(1 + \frac{1 + q/2^n}{2^n - q^2/2^n}\right) \leqslant 1 + \frac{4}{2^n}.$$

Case 4: $\boldsymbol{\tau_{i^*-j} = (x, y_0, y_1)^-_b}$, $\boldsymbol{\tau_{i^*} = (x', y'_0, y'_1)^-_b}$

As before we can assume $b = 0$. With S and S' as in Case 2 and $\mathsf{S}^\dagger$ and $\mathsf{S}'^\dagger$ as in Case 3 we can use (8) repeatedly to have

$$\mathbb{P}_{\mathrm{id}}\left[\tau_{i^*-j} \mid \tau^{(i,j)}_{\mathsf{head}}\right] = \frac{|\mathsf{S}^\dagger| + 2^n - q|\mathsf{D}|/2^n}{(2^n - |\mathsf{D}|)^2}, \tag{42}$$

$$\mathbb{P}_{\mathrm{id}}\left[\tau_{i^*} \mid \tau_{i^*-j}, \tau^{(i,j)}_{\mathsf{head}}\right] = \frac{|\mathsf{S}'| + 2^n - q|\mathsf{D} \cup \{x\}|/2^n}{(2^n - |\mathsf{D} \cup \{x\}|)^2}, \tag{43}$$

$$\mathbb{P}_{\mathrm{id}}\left[\tau_{i^*} \mid \tau^{(i,j)}_{\mathsf{head}}\right] = \frac{|\mathsf{S}| + 2^n - q|\mathsf{D}|/2^n}{(2^n - |\mathsf{D}|)^2}, \tag{44}$$

$$\mathbb{P}_{\mathrm{id}}\left[\tau_{i^*-j} \mid \tau_{i^*}, \tau^{(i,j)}_{\mathsf{head}}\right] = \frac{|\mathsf{S}'^\dagger| + 2^n - q|\mathsf{D} \cup \{x'\}|/2^n}{(2^n - |\mathsf{D} \cup \{x'\}|)^2}. \tag{45}$$

From (42)-(45) we get

$$\begin{aligned}\rho^{(i,j)}_\tau &= \frac{|\mathsf{S}^\dagger| + 2^n - q|\mathsf{D}|/2^n}{|\mathsf{S}'^\dagger| + 2^n - q(|\mathsf{D}|+1)/2^n} \cdot \frac{|\mathsf{S}'| + 2^n - q(|\mathsf{D}|+1)/2^n}{|\mathsf{S}| + 2^n - q|\mathsf{D}|/2^n} \\ &= \left(1 + \frac{|\mathsf{S}^\dagger| - |\mathsf{S}'^\dagger| + q/2^n}{|\mathsf{S}'^\dagger| + 2^n - q(|\mathsf{D}|+1)/2^n}\right)\left(1 + \frac{|\mathsf{S}'| - |\mathsf{S}| - q/2^n}{|\mathsf{S}| + 2^n - q|\mathsf{D}|/2^n}\right).\end{aligned}$$

When $|\mathsf{S}^\dagger \setminus \mathsf{S}'^\dagger| = |\mathsf{S}' \setminus \mathsf{S}| = 0$, we get the bound

$$\rho^{(i,j)}_\tau \leqslant 1 + \frac{q/2^n}{|\mathsf{S}'^\dagger| + 2^n - q(|\mathsf{D}|+1)/2^n} \leqslant 1 + \frac{2q}{2^{2n}}.$$

Recalling from cases 2 and 3 that $|\mathsf{S}^\dagger \setminus \mathsf{S}'^\dagger| > 0$ or $|\mathsf{S}' \setminus \mathsf{S}| > 0$ can only hold for erratic pairs. For these we use the bounds $|\mathsf{S}^\dagger \setminus \mathsf{S}'^\dagger| \leqslant 1$ and $|\mathsf{S}' \setminus \mathsf{S}| \leqslant 2$ to get

$$\begin{aligned}\rho^{(i,j)}_\tau &\leqslant \left(1 + \frac{1 + q/2^n}{|\mathsf{S}'^\dagger| + 2^n - q(|\mathsf{D}|+1)/2^n}\right)\left(1 + \frac{2 - q/2^n}{|\mathsf{S}| + 2^n - q|\mathsf{D}|/2^n}\right) \\ &\leqslant \left(1 + \frac{2}{2^n}\right)\left(1 + \frac{2}{2^n}\right) \leqslant 1 + \frac{5}{2^n}.\end{aligned}$$

Case 5: $\boldsymbol{\tau_{i^*-j} = (x, y_0, y_1)^-_b}$, $\boldsymbol{\tau_{i^*} = (x', y'_0, y'_1)^-_{1-b}}$

Again we assume $b = 0$. This case is very close to Case 4. The expressions for $\mathbb{P}_{\mathrm{id}}\left[\tau_{i^*-j} \mid \tau^{(i,j)}_{\mathsf{head}}\right]$ and $\mathbb{P}_{\mathrm{id}}\left[\tau_{i^*-j} \mid \tau_{i^*}, \tau^{(i,j)}_{\mathsf{head}}\right]$ carry over unchanged from (42) and (45) in Case 4. For the other two, using (8) twice yields

$$\mathbb{P}_{\mathrm{id}}\left[\tau_{i^*} \mid \tau_{i^*-j}, \tau^{(i,j)}_{\mathsf{head}}\right] = \frac{|\mathsf{S}_{\mathsf{D_C} \setminus (\mathsf{D} \cup \{x\}) \to (\mathsf{R}_0 \cup \{y_0\}) \oplus y'_1}| + 2^n - q|\mathsf{D} \cup \{x\}|/2^n}{(2^n - |\mathsf{D} \cup \{x\}|)^2}, \tag{46}$$

$$\mathbb{P}_{\mathrm{id}}\left[\tau_{i^*} \mid \tau^{(i,j)}_{\mathsf{head}}\right] = \frac{|\mathsf{S}_{\mathsf{D_C} \setminus \mathsf{D} \to \mathsf{R}_0 \oplus y'_1}| + 2^n - q|\mathsf{D}|/2^n}{(2^n - |\mathsf{D}|)^2}. \tag{47}$$

Defining $\mathsf{S}^\star := \mathsf{S}_{\mathsf{D_C}\setminus\mathsf{D}\to\mathsf{R}_0\oplus y_1'}$ and $\mathsf{S}'^\star := \mathsf{S}_{\mathsf{D_C}\setminus(\mathsf{D}\cup\{x\})\to(\mathsf{R}_0\cup\{y_0\})\oplus y_1'}$ and simplifying as in Case 4 yields

$$\rho_\tau^{(i,j)} = \left(1 + \frac{|\mathsf{S}^\dagger| - |\mathsf{S}'^\dagger| + q/2^n}{|\mathsf{S}'^\dagger| + 2^n - q(|\mathsf{D}|+1)/2^n}\right)\left(1 + \frac{|\mathsf{S}'^\star| - |\mathsf{S}^\star| - q/2^n}{|\mathsf{S}^\star| + 2^n - q|\mathsf{D}|/2^n}\right).$$

The rest of the analysis in Case 4 only uses observations on the difference $|\mathsf{S}'|-|\mathsf{S}|$, which by symmetry also hold for the difference $|\mathsf{S}'^\star|-|\mathsf{S}^\star|$. Thus the bounds from Case 4 also hold for Case 5.

Lemma 6. *Let X_1 and X_2 be random variables with means μ_1 and μ_2 and with variances σ_1^2 and σ_2^2. Then for $\mu = (\mu_1+\mu_2)/2$, the midpoint of the means, we have*

$$|\mathbb{P}[X_1 \leqslant \mu] - \mathbb{P}[X_2 \leqslant \mu]| \geqslant 1 - \frac{4(\sigma_1^2+\sigma_2^2)}{(\mu_1-\mu_2)^2}.$$

Proof. Assume without loss of generality that $\mu_1 < \mu_2$ so that $\mu_2 - \mu = \mu - \mu_1 = (\mu_2-\mu_1)/2$. We will show that

$$\mathbb{P}[X_1 \leqslant \mu] \geqslant 1 - \frac{4\sigma_1^2}{(\mu_1-\mu_2)^2} \tag{48}$$

$$\mathbb{P}[X_2 \leqslant \mu] \leqslant \frac{4\sigma_2^2}{(\mu_1-\mu_2)^2}, \tag{49}$$

which leads to

$$\begin{aligned}|\mathbb{P}[X_1 \leqslant \mu] - \mathbb{P}[X_2 \leqslant \mu]| &\geqslant 1 - \frac{4\sigma_1^2}{(\mu_1-\mu_2)^2} - \frac{4\sigma_2^2}{(\mu_1-\mu_2)^2} \\ &= 1 - \frac{4(\sigma_1^2+\sigma_2^2)}{(\mu_1-\mu_2)^2},\end{aligned}$$

as desired.

We are left to prove (48) and (49), starting with (48) we derive

$$\begin{aligned}\sigma_1^2 = \mathbb{E}\left[(X_1-\mu_1)^2\right] &= \sum_x \mathbb{P}[X_1 = x]\cdot(x-\mu_1)^2 \\ &\geqslant \sum_{x>\mu} \mathbb{P}[X_1 = x]\cdot((x-\mu)+(\mu-\mu_1))^2.\end{aligned}$$

As both $x-\mu > 0$ and $\mu - \mu_1 = (\mu_2-\mu_1)/2 > 0$ we get

$$\sigma_1^2 \geqslant \sum_{x>\mu} \mathbb{P}[X_1 = x]\cdot\frac{(\mu_2-\mu_1)^2}{4} = \mathbb{P}[X_1 > \mu]\cdot\frac{(\mu_1-\mu_2)^2}{4},$$

resulting in

$$\mathbb{P}[X_1 \leqslant \mu] = 1 - \mathbb{P}[X_1 > \mu] \geqslant \frac{4\sigma_1^2}{(\mu_1-\mu_2)^2}$$

as desired. Similarly, for (49) we get that

$$\sigma_2^2 \geqslant \sum_{x \leqslant \mu} \mathbb{P}\left[X_2 = x\right] \cdot ((\mu_2 - \mu) + (\mu - x))^2 \geqslant \mathbb{P}\left[X_2 \leqslant \mu\right] \cdot \frac{(\mu_2 - \mu_1)^2}{4},$$

resulting in

$$\mathbb{P}\left[X_2 \leqslant \mu\right] \leqslant \frac{4\sigma_2^2}{4(\mu_1 - \mu_2)^2}$$

as desired, finishing the proof.

Lemma 7. *Using the notation in Section 6.3, we have*

$$\delta_j \geqslant \frac{q^3}{2^{3n+2}} + \mathcal{O}\left(\frac{q^4}{2^{4n}}\right).$$

Proof. The values V_j and W_j depend on the output of the random oracle z_j. Although its input x_j is fresh from the perspective of the distinguisher, it is not necessarily uniformly distributed. It is the same problem as described in Section 3.2: the inverse simulator might have queried the random oracle for the same x_j but rejected it. Because of this, we separate between a fresh, hence uniform, random oracle query and a previously queried input for which we do not make any assumptions:

$$\begin{aligned} p_j &= \mathbb{E}\left[\frac{q - W_j}{2^n - V_j} \mid x_j \text{ new}\right] \mathbb{P}\left[x_j \text{ new}\right] + \mathbb{E}\left[\frac{q - W_j}{2^n - V_j} \mid x_j \text{ old}\right] \mathbb{P}\left[x_j \text{ old}\right] \\ &= \mathbb{E}_j^{\text{new}}\left[\frac{q - W_j}{2^n - V_j}\right] \mathbb{P}\left[x_j \text{ new}\right] + \mathbb{E}_j^{\text{old}}\left[\frac{q - W_j}{2^n - V_j}\right] \mathbb{P}\left[x_j \text{ old}\right], \end{aligned}$$

where we define $\mathbb{E}_j^{\text{new}}\left[\cdot\right] = \mathbb{E}\left[\cdot \mid x_j \text{ new}\right]$ as a shorthand notation and similar for 'old'. Let

$$\begin{aligned} \delta_j^{\text{new}} &= \mathbb{E}_j^{\text{new}}\left[\frac{q - W_{I,j}}{2^n - V_{I,j}} - \frac{q - W_{\varnothing,j}}{2^n - V_{\varnothing,j}}\right], \\ \delta_j^{\text{old}} &= \mathbb{E}_j^{\text{old}}\left[\frac{q - W_{I,j}}{2^n - V_{I,j}} - \frac{q - W_{\varnothing,j}}{2^n - V_{\varnothing,j}}\right], \end{aligned}$$

then

$$\delta_j = \delta_j^{\text{new}} \cdot \mathbb{P}\left[x_j \text{ new}\right] + \delta_j^{\text{old}} \cdot \mathbb{P}\left[x_j \text{ old}\right].$$

First we compute simple bounds for $\mathbb{P}\left[x_j \text{ new}\right]$ and $\mathbb{P}\left[x_j \text{ old}\right]$. Assume that the inverse simulator tries an arbitrary ℓ times to get a value, then the expected number of additional rejected values for a single inverse query is at most

$$\sum_{i=1}^{\ell-1} \left(\frac{q}{2^n}\right)^i \leqslant \sum_{i=1}^{\infty} \left(\frac{q}{2^n}\right)^i = \frac{q}{2^n} \frac{1}{1 - q/2^n} = \frac{q}{2^n - q}.$$

As there are q inverse queries, the total expected number of values is at most

$$\frac{q^2}{2^n - q}.$$

Therefore, the probability that a fresh query after the inverse queries was previously rejected is at most

$$\mathbb{P}\left[x_j \text{ old}\right] \leqslant \frac{q^2}{(2^n - q)^2} \leqslant \frac{2q^2}{2^{2n}},$$

as $q \leqslant 2^{n-2}$ and there are $2^n - q$ possibilities left to choose from. This also implies that

$$\mathbb{P}\left[x_j \text{ new}\right] \geqslant 1 - \frac{2q^2}{2^{2n}} \geqslant \frac{1}{2}.$$

From this we also get the following upper and lower bound for $|I|$:

$$\begin{aligned}
\mathbb{E}\left[|I|\right] &= \sum_i \mathbb{P}\left[\lfloor z_i \rfloor_{n-k} = 0^{n-k}\right] \\
&= \sum_i \mathbb{P}\left[\lfloor z_i \rfloor_{n-k} = 0^{n-k} \mid x_i \text{ new}\right] \cdot \mathbb{P}\left[x_i \text{ new}\right] \\
&+ \sum_i \mathbb{P}\left[\lfloor z_i \rfloor_{n-k} = 0^{n-k} \mid x_i \text{ old}\right] \cdot \mathbb{P}\left[x_i \text{ old}\right] \\
&\leqslant \sum_i \frac{q}{2^n} \cdot 1 + \sum_i 1 \cdot \frac{2q^2}{2^{2n}} \leqslant \frac{2q^2}{2^n}, \\
\mathbb{E}\left[|I|\right] &= \sum_i \mathbb{P}\left[\lfloor z_i \rfloor_{n-k} = 0^{n-k}\right] \\
&\geqslant \sum_i \mathbb{P}_{\text{new}}\left[\lfloor z_i \rfloor_{n-k} = 0^{n-k}\right] \cdot \mathbb{P}\left[x_i \text{ new}\right] \\
&\geqslant \sum_i \frac{q}{2^n} \cdot \frac{1}{2} = \frac{q^2}{2^{n+1}},
\end{aligned}$$

using that $q \leqslant 2^{n-1}$. We will now focus on computing upper bounds on δ_j^{new} and δ_j^{old}. From Lemma 11 of the Supplementary Material we get that

$$\delta_j^{\text{new}} \geqslant \frac{q^3}{2^{3n+1}} + \mathcal{O}\left(\frac{q^4}{2^{4n}}\right).$$

Now we look at δ_j^{old} where the simulator already queried x_j to the random oracle, which means that we cannot assume that z_j is uniformly distributed. While only a few values are rejected, it is sufficient for us to not assume any structure for it. We consider

$$\delta_j^{\text{old}} = \mathbb{E}_j^{\text{old}}\left[\frac{q - W_{I,j}}{2^n - V_{I,j}} - \frac{q - W_{\varnothing,j}}{2^n - V_{\varnothing,j}}\right]$$

and we know in general that $W_{I,j} \leqslant W_{\varnothing,j} + |I|$ and $V_{I,j} \geqslant V_{\varnothing,j}$, hence

$$\begin{aligned}\delta_j^{\text{old}} &\geqslant \mathbb{E}_j^{\text{old}}\left[\frac{q - W_{\varnothing,j} - |I|}{2^n - V_{\varnothing,j}} - \frac{q - W_{\varnothing,j}}{2^n - V_{\varnothing,j}}\right] \\ &= \mathbb{E}_j^{\text{old}}\left[\frac{-|I|}{2^n - V_{\varnothing,j}}\right] \geqslant \mathbb{E}_j^{\text{old}}\left[\frac{-2|I|}{2^n}\right] \geqslant \frac{-4q^2}{2^{2n}},\end{aligned}$$

using that $V_{\varnothing,j} \leqslant 2(q + (j-1)) \leqslant 4q$ and $q \leqslant 2^{n-3}$.

Putting this together we get

$$\begin{aligned}\delta_j &= \delta_j^{\text{new}} \cdot \mathbb{P}\left[x_j \text{ new}\right] + \delta_j^{\text{old}} \cdot \mathbb{P}\left[x_j \text{ old}\right] \\ &\geqslant \left(\frac{q^3}{2^{3n+1}} + \mathcal{O}\left(\frac{q^4}{2^{4n}}\right)\right) \cdot \frac{1}{2} - \frac{4q^2}{2^{2n}} \cdot \frac{2q^2}{2^{2n}} = \frac{q^3}{2^{3n+2}} + \mathcal{O}\left(\frac{q^4}{2^{4n}}\right).\end{aligned}$$

Lemma 8. *Using the notation in Section 6.3 and Lemma 7, we have the following expectations of $V_{\varnothing,j}$ and $V_{I,j}$:*

$$\begin{aligned}\mathbb{E}_j^{\text{new}}\left[V_{\varnothing,j}\right] &= 2(q + (j-1)) - \frac{(q + (j-1))^2}{2^n}, \\ \mathbb{E}_j^{\text{new}}\left[V_{I,j}\right] &= 2(q + |I| + (j-1)) - \frac{(q + |I| + (j-1))^2}{2^n}.\end{aligned}$$

For $W_{\varnothing,j}$ and $W_{I,j}$ we have:

$$\begin{aligned}\mathbb{E}_j^{\text{new}}\left[W_{\varnothing,j}\right] &= \left(\frac{q(2q + (j-1))}{2^n} + p_{\varnothing,j-1}^{\text{total}}\right) - \left(\frac{q^2}{2^n} + p_{\varnothing,j-1}^{\text{total}}\right)\frac{q + (j-1)}{2^n}, \\ \mathbb{E}_j^{\text{new}}\left[W_{I,j}\right] &= \left(\frac{q(2q + |I| + (j-1))}{2^n} + p_{I,j-1}^{\text{total}}\right) - \left(\frac{q^2}{2^n} + p_{I,j-1}^{\text{total}}\right)\frac{q + |I| + (j-1)}{2^n},\end{aligned}$$

where

$$p_{\varnothing,j-1}^{\text{total}} = \sum_{j' \leqslant j-1} p_{\varnothing,j'}, \qquad p_{I,j-1}^{\text{total}} = \sum_{j' \leqslant j-1} p_{I,j'}.$$

Proof. As we assume that we have an input that has not been queried (implicitly by the simulator) to the random oracle, we can assume that z_j is uniformly drawn from $\{0,1\}^n$. Let r denote the number of previous queries (the sizes of R_0 and R_1 in the simulator). As z_j is uniform, the expected value of V is $2r - r^2/2^n$. The $2r$ comes from the fact that all values in the ranges are excluded, and the $-r^2/2^n$ corrects double counting.

The expected value of W_j depends on the kind of previous queries. For permutation 0 a possibility is excluded when $\lfloor y \rfloor_{n-k} = 0^{n-k}$ for the particular y, while for permutation 1 this happens when $\lfloor y' \oplus z_j \rfloor_{n-k} = 0^{n-k}$ for the particular y'. As z_j is uniformly sampled, the probability for permutation 1 is always $q/2^n$ for a single query. This results in the following contributions for the different query types.

Inverse query $\mathcal{S}_1^{-1}(0^{n-k} \parallel y)$

- Permutation 0: the simulator implicitly sets the value according to the random oracle. While there can be some biases in the distribution of this set, the probability for a single element is still $q/2^n$ as no single value is more likely than any other, giving an expected number of $q^2/2^n$;
- Permutation 1: as the probability for a single query is always $q/2^n$, we get an expectation of $q^2/2^n$ for q queries.

Queries in I

- Permutation 0: as for $i \in I$ we know that $\lfloor z_i \rfloor_{n-k} = 0^{n-k}$ we cannot have that $\lfloor y_i \rfloor_{n-k} = 0^{n-k}$ hence this value is always 0. This is the core of the attack;
- Permutation 1: for $|I|$ queries we get an expectation of $|I|q/2^n$.

Earlier forward queries

- Permutation 0: the probability that a specific $j' < j$ satisfies $\lfloor y_{j'} \rfloor_{n-k} = 0^{n-k}$ is $p_{j'}$ (which is roughly $q/2^n$), giving an expected number of $p_{j-1}^{\text{total}} = \sum_{j' \leqslant j-1} p_{j'}$;
- Permutation 1: for $j-1$ queries we get an expectation of $(j-1)q/2^n$.

Let $W_{\varnothing,j}$ and $V_{\varnothing,j}$ denote these values for when the queries I are skipped and let $W_{I,j}$ and $V_{I,j}$ denote these values for when the queries I are included. We conclude that

$$\mathbb{E}_j^{\text{new}}\left[V_{\varnothing,j}\right] = 2(q+(j-1)) - \frac{(q+(j-1))^2}{2^n},$$
$$\mathbb{E}_j^{\text{new}}\left[V_{I,j}\right] = 2(q+|I|+(j-1)) - \frac{(q+|I|+(j-1))^2}{2^n},$$

as there are $q+(j-1)$ queries made in the first case and $q+|I|+(j-1)$ in the second case.

For W we have to be a bit more careful about correcting for the double counting as we have to split r into r_0 and r_1 for permutation 0 and permutation 1 (which can be different), resulting in a correction of $-r_0 r_1/q$. This results in

$$\mathbb{E}_j^{\text{new}}\left[W_{\varnothing,j}\right] = \left(\frac{q(2q+(j-1))}{2^n} + p_{\varnothing,j-1}^{\text{total}}\right) - \left(\frac{q^2}{2^n} + p_{\varnothing,j-1}^{\text{total}}\right)\frac{q+(j-1)}{2^n},$$
$$\mathbb{E}_j^{\text{new}}\left[W_{I,j}\right] = \left(\frac{q(2q+|I|+(j-1))}{2^n} + p_{I,j-1}^{\text{total}}\right) - \left(\frac{q^2}{2^n} + p_{I,j-1}^{\text{total}}\right)\frac{q+|I|+(j-1)}{2^n},$$

as the expected exclusions from permutation 0 is $q^2/2^n + p_{j-1}^{\text{total}}$ in both cases (with the last term specified per case) and the expected exclusions from permutation 1 is $q/2^n$ multiplied by the total number of queries made, already established at $q+(j-1)$ or $q+|I|+(j-1)$. Note that exclusions from permutation 1 is independent of the value given by permutation 0, making the correction term for double counting work.

Lemma 9. *Using the notation in Section 6.3, the following basic bounds on $p_{\varnothing,j}$ and $p_{I,j}$ hold for all j:*

$$p_{\varnothing,j}, p_{I,j} = \frac{q}{2^n} + \mathcal{O}\left(\frac{q^3}{2^{3n}}\right).$$

Proof. We use strong induction on j. We focus on $p_{I,j}$ as the case of $p_{\varnothing,j}$ analogous with $|I|$ removed. As big O notation does not work well with induction, we have to provide an explicit error term. For this, we first look at Lemma 8, from which we can derive that

$$q\mathbb{E}_j^{\text{new}}[V_{I,j}] = 2q(q + |I| + (j-1)) + \mathcal{O}\left(\frac{q^3}{2^n}\right),$$
$$2^n\mathbb{E}_j^{\text{new}}[W_{I,j}] = q(2q + |I| + (j-1)) + 2^n p_{I,j-1}^{\text{total}} + \mathcal{O}\left(\frac{q^3}{2^n}\right),$$

hence, using that $\mathbb{E}[|I|] \in \mathcal{O}\left(q^2/2^n\right)$,

$$q\mathbb{E}_j^{\text{new}}[V_{I,j}] - 2^n\mathbb{E}_j^{\text{new}}[W_{I,j}] = (j-1)q - 2^n p_{I,j-1}^{\text{total}} + g(q,n), \tag{50}$$

for some g with $|g(q,n)| \leqslant C_1 q^3/2^n$. We will show that

$$p_{I,j} = \frac{q}{2^n} + f(q,n)$$

for some f with $|f(q,n)| \leqslant Cq^3/2^{3n}$ for some to be determined constant C. From the induction hypothesis we immediately derive that

$$\begin{aligned} p_{I,j-1}^{\text{total}} &= \sum_{j' \leqslant j-1} p_{I,j'} = \sum_{j' \leqslant j-1} \left(\frac{q}{2^n} + f(q,n)\right) \\ &= \frac{(j-1)q}{2^n} + (j-1)f(q,n). \end{aligned}$$

Now we derive

$$\begin{aligned} \mathbb{E}_j^{\text{new}}\left[\frac{q - W_{I,j}}{2^n - V_{I,j}}\right] &= \frac{q}{2^n} + \mathbb{E}_j^{\text{new}}\left[\frac{q - W_{I,j}}{2^n - V_{I,j}} - \frac{q}{2^n}\right] \\ &= \frac{q}{2^n} + \mathbb{E}_j^{\text{new}}\left[\frac{q2^n - 2^n W_{I,j} - q2^n + qV_{I,j}}{2^n(2^n - V_{I,j})}\right] \\ &= \frac{q}{2^n} + \frac{h(q,n)}{2^{2n}}\left(q\mathbb{E}_j^{\text{new}}[V_{I,j}] - 2^n\mathbb{E}_j^{\text{new}}[W_{I,j}]\right), \end{aligned}$$

where $|h(q,n)| \leqslant 4$, using that $V_{I,j} \leqslant 2(q + |I| + (j-1)) \leqslant 6q$ and $q \leqslant 2^{n-3}$. Using (50) we continue with

$$\begin{aligned} \mathbb{E}_j^{\text{new}}\left[\frac{q - W_{I,j}}{2^n - V_{I,j}}\right] &= \frac{q}{2^n} + \frac{h(q,n)}{2^{2n}}\left((j-1)q - 2^n p_{I,j-1}^{\text{total}} + g(q,n)\right) \\ &= \frac{q}{2^n} + \frac{h(q,n)}{2^{2n}}\left(g(q,n) - (j-1)f(q,n)\right) \end{aligned}$$

$$= \frac{q}{2^n} + f'(q,n),$$

where for $f'(q,n) = h(q,n)(g(q,n) - (j-1)f(q,n))/2^{2n}$ we have that

$$\begin{aligned}
|f'(q,n)| &\leqslant \frac{|h(q,n)|}{2^{2n}} \left(|g(q,n)| + (j-1)|f(q,n)|\right) \\
&\leqslant \frac{4}{2^{2n}} \left(\frac{C_1 q^3}{2^n} + \frac{qCq^3}{2^{3n}}\right) \leqslant \frac{4C_1 q^3}{2^{3n}} + \frac{4Cq^3}{2^{4n+3}} \\
&= \left(4C_1 + \frac{4C}{2^{n+3}}\right) \frac{q^3}{2^{3n}},
\end{aligned}$$

using that $q \leqslant 2^{n-3}$. We know that

$$\begin{aligned}
p_{I,j} &= \mathbb{E}_j^{\text{new}} \left[\frac{q - W_{I,j}}{2^n - V_{I,j}}\right] \mathbb{P}\left[x_j \text{ new}\right] + \mathbb{E}_j^{\text{old}} \left[\frac{q - W_{I,j}}{2^n - V_{I,j}}\right] \mathbb{P}\left[x_j \text{ old}\right] \\
&= \frac{q}{2^n} \mathbb{P}\left[x_j \text{ new}\right] + \mathbb{E}_j^{\text{old}} \left[\frac{q - W_{I,j}}{2^n - V_{I,j}}\right] \mathbb{P}\left[x_j \text{ old}\right] + f''(q,n),
\end{aligned}$$

with $f''(q,n) = \mathbb{P}\left[x_j \text{ new}\right] f'(q,n)$, so $|f''(q,n)| \leqslant |f'(q,n)|$. We derive the following two bounds on $p_{I,j}$ to get the desired result

$$\begin{aligned}
p_{I,j} &\leqslant \frac{q}{2^n} + \mathbb{E}_j^{\text{old}} \left[\frac{q - W_{I,j}}{2^n - V_{I,j}}\right] \frac{2q^2}{2^{2n}} + f''(q,n) \\
&\leqslant \frac{q}{2^n} + \mathbb{E}_j^{\text{old}} \left[\frac{1}{2^n - V_{I,j}}\right] \frac{2q^3}{2^{2n}} + f''(q,n) \leqslant \frac{q}{2^n} + \frac{8q^3}{2^{3n}} + f''(q,n), \\
p_{I,j} &\geqslant \frac{q}{2^n}\left(1 - \frac{2q^2}{2^{2n}}\right) + f''(q,n) = \frac{q}{2^n} - \frac{2q^3}{2^{3n}} + f''(q,n),
\end{aligned}$$

again using that $V_{I,j} \leqslant 2(q + |I| + (j-1)) \leqslant 6q$ and $q \leqslant 2^{n-3}$. We can write this as

$$p_{I,j} = \frac{q}{2^n} + e(q,n),$$

where we have to show that $|e(q,n)| \leqslant Cq^3/2^{3n}$. In both cases we get that

$$|e(q,n)| \leqslant \frac{8q^3}{2^{3n}} + |f'(q,n)| \leqslant \left(8 + 4C_1 + \frac{4C}{2^{n+3}}\right) \frac{q^3}{2^{3n}} \leqslant \left(8 + 4C_1 + \frac{C}{2}\right) \frac{q^3}{2^{3n}}.$$

Let $C = 16 + 8C_1$, then

$$|e(q,n)| \leqslant \left(\frac{C}{2} + \frac{C}{2}\right) \frac{q^3}{2^{3n}} = \frac{Cq^3}{2^{3n}},$$

as desired.

Lemma 10. *For any arbitrary numbers a, a', b, b' we have*

$$ab - a'b' = \frac{(a+a')(b-b') + (a-a')(b+b')}{2}.$$

Proof. By writing out the right hand side we simply get

$$\begin{aligned}&\frac{(a+a')(b-b')+(a-a')(b+b')}{2}\\&=\frac{(ab-ab'+a'b-a'b')+(ab+ab'-a'b-a'b')}{2}\\&=\frac{2ab-2a'b'}{2}=ab-a'b',\end{aligned}$$

which is the left hand side.

Lemma 11. *Using the notation in Lemma 7, we have*

$$\delta_j^{\text{new}} \geqslant \frac{q^3}{2^{3n+1}}+\mathcal{O}\left(\frac{q^4}{2^{4n}}\right).$$

Proof. From Lemma 8 we know that

$$\begin{aligned}\mathbb{E}_j^{\text{new}}\left[V_{\varnothing,j}\right]&=2(q+(j-1))-\frac{(q+(j-1))^2}{2^n},\\\mathbb{E}_j^{\text{new}}\left[V_{I,j}\right]&=2(q+|I|+(j-1))-\frac{(q+|I|+(j-1))^2}{2^n},\end{aligned}$$

which means that

$$\mathbb{E}_j^{\text{new}}\left[V_{I,j}-V_{\varnothing,j}\right]=2\mathbb{E}\left[|I|\right]+\mathcal{O}\left(\frac{q^3}{2^{2n}}\right).$$

Additionally, we know that

$$\begin{aligned}\mathbb{E}_j^{\text{new}}\left[W_{\varnothing,j}\right]&=\left(\frac{q(2q+(j-1))}{2^n}+p_{\varnothing,j-1}^{\text{total}}\right)-\left(\frac{q^2}{2^n}+p_{\varnothing,j-1}^{\text{total}}\right)\frac{q+(j-1)}{2^n},\\\mathbb{E}_j^{\text{new}}\left[W_{I,j}\right]&=\left(\frac{q(2q+|I|+(j-1))}{2^n}+p_{I,j-1}^{\text{total}}\right)-\left(\frac{q^2}{2^n}+p_{I,j-1}^{\text{total}}\right)\frac{q+|I|+(j-1)}{2^n}.\end{aligned}$$

Combining this with Lemma 9, which states that

$$\begin{aligned}p_{\varnothing,j-1}^{\text{total}}&=\sum_{j'\leqslant j-1}p_{\varnothing,j'}=\frac{(j-1)q}{2^n}+\mathcal{O}\left(\frac{q^4}{2^{3n}}\right),\\p_{I,j-1}^{\text{total}}&=\sum_{j'\leqslant j-1}p_{I,j'}=\frac{(j-1)q}{2^n}+\mathcal{O}\left(\frac{q^4}{2^{3n}}\right),\end{aligned}$$

we can simplify to

$$\begin{aligned}\mathbb{E}_j^{\text{new}}\left[W_{\varnothing,j}\right]&=\frac{q(2q+2(j-1))}{2^n}-\frac{q(q+(j-1))^2}{2^{2n}}+\mathcal{O}\left(\frac{q^4}{2^{3n}}\right),\\\mathbb{E}_j^{\text{new}}\left[W_{I,j}\right]&=\frac{q(2q+|I|+2(j-1))}{2^n}-\frac{q(q+(j-1))(q+|I|+(j-1))}{2^{2n}}+\mathcal{O}\left(\frac{q^4}{2^{3n}}\right),\end{aligned}$$

concluding that

$$\mathbb{E}_j^{\text{new}}\left[W_{I,j} - W_{\varnothing,j}\right] = \frac{q}{2^n}\mathbb{E}\left[|I|\right] + \mathcal{O}\left(\frac{q^4}{2^{3n}}\right).$$

For the difference in the probabilities for a single query j we get

$$\begin{aligned}
\delta_j^{\text{new}} &= \mathbb{E}_j^{\text{new}}\left[\frac{q - W_{I,j}}{2^n - V_{I,j}} - \frac{q - W_{\varnothing,j}}{2^n - V_{\varnothing,j}}\right] \\
&= \mathbb{E}_j^{\text{new}}\left[\frac{(q - W_{I,j})(2^n - V_{\varnothing,j}) - (q - W_{\varnothing,j})(2^n - V_{I,j})}{(2^n - V_{I,j})(2^n - V_{\varnothing,j})}\right] \\
&\geqslant \frac{1}{2^{2n}}\mathbb{E}_j^{\text{new}}\left[q2^n - qV_{\varnothing,j} - 2^n W_{I,j} + W_{I,j}V_{\varnothing,j} - q2^n + qV_{I,j} + 2^n W_{\varnothing,j} - W_{\varnothing,j}V_{I,j}\right] \\
&= \frac{1}{2^{2n}}\mathbb{E}_j^{\text{new}}\left[q(V_{I,j} - V_{\varnothing,j}) - 2^n(W_{I,j} - W_{\varnothing,j}) + W_{I,j}V_{\varnothing,j} - W_{\varnothing,j}V_{I,j}\right] \\
&= \frac{1}{2^{2n}}\left(q\mathbb{E}_j^{\text{new}}\left[V_{I,j} - V_{\varnothing,j}\right] - 2^n\mathbb{E}_j^{\text{new}}\left[W_{I,j} - W_{\varnothing,j}\right] + \mathbb{E}_j^{\text{new}}\left[W_{I,j}V_{\varnothing,j} - W_{\varnothing,j}V_{I,j}\right]\right).
\end{aligned}$$

Using the known expectations of the differences we continue with

$$\begin{aligned}
\delta_j^{\text{new}} &\geqslant \frac{1}{2^{2n}}\left(2q\mathbb{E}\left[|I|\right] - q\mathbb{E}\left[|I|\right] + \mathbb{E}_j^{\text{new}}\left[W_{I,j}V_{\varnothing,j} - W_{\varnothing,j}V_{I,j}\right] + \mathcal{O}\left(\frac{q^4}{2^{2n}}\right)\right) \\
&= \frac{q}{2^{2n}}\mathbb{E}\left[|I|\right] + \frac{1}{2^{2n}}\mathbb{E}_j^{\text{new}}\left[W_{I,j}V_{\varnothing,j} - W_{\varnothing,j}V_{I,j}\right] + \mathcal{O}\left(\frac{q^4}{2^{4n}}\right).
\end{aligned}$$

For $W_{I,j}V_{\varnothing,j} - W_{\varnothing,j}V_{I,j}$ we use Lemma 10, which says that

$$\begin{aligned}
\mathbb{E}_j^{\text{new}}\left[W_{I,j}V_{\varnothing,j} - W_{\varnothing,j}V_{I,j}\right] = \frac{1}{2}\mathbb{E}_j^{\text{new}}\left[(W_{I,j} + W_{\varnothing,j})(V_{\varnothing,j} - V_{I,j})\right] \\
+ \frac{1}{2}\mathbb{E}_j^{\text{new}}\left[(W_{I,j} - W_{\varnothing,j})(V_{\varnothing,j} + V_{I,j})\right].
\end{aligned}$$

Now we use that $V_{I,j} - V_{\varnothing,j} \leqslant 2|I|$ and $V_{\varnothing,j}, V_{I,j} \leqslant 6q = \mathcal{O}(q)$ (unconditional, not only in the expectation). This leads to

$$\begin{aligned}
\mathbb{E}_j^{\text{new}}\left[W_{I,j}V_{\varnothing,j} - W_{\varnothing,j}V_{I,j} \mid |I|\right] &= 2|I|\mathbb{E}_j^{\text{new}}\left[W_{\varnothing,j} + W_{I,j}\right] + \mathcal{O}(q)\,\mathbb{E}_j^{\text{new}}\left[W_{I,j} - W_{\varnothing,j}\right] \\
&= 2|I|\mathcal{O}\left(\frac{q^2}{2^n}\right) + \mathcal{O}\left(\frac{q^4}{2^{2n}}\right),
\end{aligned}$$

which combined with the law of total expectation gives

$$\mathbb{E}_j^{\text{new}}\left[W_{I,j}V_{\varnothing,j} - W_{\varnothing,j}V_{I,j}\right] = 2\mathbb{E}\left[|I|\right]\mathcal{O}\left(\frac{q^2}{2^n}\right) + \mathcal{O}\left(\frac{q^4}{2^{2n}}\right) = \mathcal{O}\left(\frac{q^4}{2^{2n}}\right),$$

meaning that this term simply falls into the big O term, so

$$\delta_j^{\text{new}} \geqslant \frac{|I|q}{2^{2n}} + \mathcal{O}\left(\frac{q^4}{2^{4n}}\right) \geqslant \frac{q^2}{2^{3n+1}} + \mathcal{O}\left(\frac{q^4}{2^{4n}}\right).$$