Sequential Indifferentiability of STH and EDM

Nilanjan Datta[1,2], Avijit Dutta[1,2], Sougata Mandal[1,3] and Hrithik Nandi[1,3]

[1] Institute for Advancing Intelligence (IAI), TCG CREST, Kolkata, India
[2] Academy of Scientific and Innovative Research (AcSIR), Ghaziabad, India
[3] Ramakrishna Mission Vivekananda Educational and Research Institute, Belur, India

Abstract. The notion of indifferentiability was proposed by Maurer et al. to bound the distinguishing advantage of a construction built on a public primitive, from a public random function. In Indocrypt'10, Mandal et al. have shown that the sum of two independent permutations is indifferentiable from a public random function up to $2^{2n/3}$ queries. Later in ACNS'15, Mennink and Preneel identified an analytical flaw of Mandal et al's result and revised the security bound to $2^{2n/3}/n$. In Eurocrypt'18, Bhattacharya and Nandi have improved their indifferentiable bound to 2^n queries, which was again identified as incorrect in the analysis by Gunsing et al. In this paper, we study the indifferentiability of a few other PRF constructions, namely STH and EDM constructions. We will show that neither STH nor STH2 is indifferentiable, which led us to propose a generalized version called gSTH. We have shown that gSTH achieves a tight l-bit security bound, where l denotes the size of the constants in terms of bits used in the construction. While we show that EDM achieves a tight $n/2$-bit indifferentiable bound with respect to our proposed simulator, single-keyed EDM is not indifferentiable from a public random function. We would like to mention that all the proofs and the attacks have been done in the sequential indifferentiability model.

Keywords: Sequential Indifferentiabilty · EDM · STH · SoP

1 Introduction

Pseudo-Random Function (PRF) is a fundamental building block of symmetric-key cryptography. Since the advent of the GGM construction [GGM84], the first provably secure PRF, numerous modern cryptographic protocols have been built upon PRFs, with their security relying entirely on the pseudorandomness of the underlying function. Despite their wide application, designing dedicated PRFs remains challenging, primarily due to the difficulty of constructing secure, non-invertible round functions that can be iterated multiple times. In contrast, Pseudo-Random Permutations (PRPs), also known as block ciphers, are more prevalent in practice and form the foundation of symmetric-key cryptography. Two primary design paradigms for block ciphers are the Feistel network and the Substitution-Permutation Network (SPN). Many well-known block ciphers, including AES [DR98], PRINCE [BCG+12], PRESENT [BKL+07], and GIFT [BPP+17], follow the SPN approach. Compared to PRFs, PRPs are easier to construct, largely because invertible round functions are simpler to design and can be efficiently composed to produce secure permutations. It was soon realized that designing PRFs from PRPs is not only practical, given the relative ease of designing PRPs, but also necessary for many cryptographic

E-mail: nilanjan.datta@tcgcrest.org (Nilanjan Datta), avirocks.dutta13@gmail.com (Avijit Dutta), sougata.mandal@tcgcrest.org (Sougata Mandal), hrithik.nandi.85@tcgcrest.org (Hrithik Nandi)

This work is licensed under a "CC BY 4.0" license.
Date of this document: 2025-08-23.

applications where PRF security is the primary goal [BKR98]. In particular, numerous cryptographic modes of operation, such as [Iwa06, WC81, CDNY18, DDNP18, DDNY18], do not require the invertibility of their underlying primitives. For these, achieving PRF security is both sufficient and more economical. Notably, counter mode encryption instantiated with a PRF provides stronger security guarantees than when instantiated with a PRP, further justifying the use of PRFs as the core primitive in such settings.

An n-bit PRP can be treated as a PRF for up to $2^{n/2}$ queries, due to the well-known PRP-PRF switching lemma [CN08]. However, this bound is inadequate when the block size of a construction is small. This is particularly relevant in the scenario when one instantiates cryptographic schemes using lightweight block ciphers like PRESENT [BKL+07], GIFT [BPP+17], whose block size is 64 bits. To address this, Hall et al. [HWKS98] proposed the Truncated Random Permutation (TRP), which generates a secure PRF by truncating an n-bit permutation to a bits. The TRP remains secure for up to $2^{n-a/2}$ queries [BI99, GG21], making it suitable when some output length reduction is acceptable. An alternative and widely studied solution is the Sum of Permutations (SoP) construction by Bellare et al. [BKR98], which combines the outputs of two independent PRPs using XOR. This yields a secure PRF withstanding up to 2^n queries [DHT17, DNS22]. Over the past decade, several other PRP-based PRF constructions have been developed, including EDM [CS16a], Dual EDM (EDMD) [MN17], SUMPIP [GSWG19], and the Summation-Truncation Hybrid (STH) [GM20]. All these constructions achieve security beyond the $2^{n/2}$ query bound, offering practical solutions in contexts where standard PRF designs are either infeasible or inefficient.

1.1 Indifferentiability Framework

The security results of the above constructions are in the indistinguishability framework, where the underlying primitive has been assumed to be a PRP, and the bound captures the advantage of the best possible adversary in distinguishing the output of the construction from a random function. A naturally related question is to what degree the constructions behave like a public random function when they are instantiated with public permutations. As the constructions are now keyless (as they are instantiated with public permutations, not with *secret keyed* permutations), we cannot apply the conventional indistinguishability security bound. Instead, we should consider the security analysis of these constructions in the *indifferentiability* framework. The notion of indifferentiability was first introduced by Maurer et al. [MRH04], and later adapted to the context of symmetric cryptographic primitives by Coron et al. [CDMP05]. Informally, a construction C using an ideal primitive P is said to be indifferentiable from another ideal primitive F if there exists a simulator $\mathcal{S}$, with access to F, such that $(\mathsf{C}^{\mathsf{P}}, \mathsf{P})$ and $(\mathsf{F}, \mathcal{S}^{\mathsf{F}})$ are indistinguishable. Broadly speaking, the simulator aims to answer queries in a way that they remain consistent with F, while staying close to the distribution of P's responses. Indifferentiability supports modular security proofs and can be seen as a specific instance of the Universal Composition (UC) framework. In simple terms, if C^{P} is indifferentiable from F, then any cryptosystem secure when using F in a single-stage setting [RSS11] remains secure when F is replaced with C^{P}.

In the context of the indifferentiable security of the Feistel construction, Mandal et al. [MPS12] introduced a weaker notion known as *sequential indifferentiability*. In this model, the distinguisher must make all its queries to the ideal primitive P (or the simulator $\mathcal{S}$) before querying the construction C^{P} (or the ideal primitive F). Although this notion is weaker than regular indifferentiability, it is relatively easier to use in practice and has been effectively applied to several constructions, including the 6-round Feistel network [MPS12], the $(r+3)$-round tweakable Even-Mansour cipher with linear tweak and key mixing [CEJS23], and the SoP construction [MPN10, MP15, BN18, Lee17]. Furthermore, sequential indifferentiability has been shown to be strong enough to imply the correlation

intractability of a cipher [MPS12].

1.2 Indifferentiability Results for PRP-Based PRFs

In this section, we revisit the existing results on indifferentiable security of some well-known PRP-based PRF constructions.

Indifferentiability for TRP. The Indifferentiability of the Truncated Random Permutation was first analyzed by Dodis et al. [DRRS09], who showed that TRP is indifferentiable from a public random function up to the following bound:

$$\frac{(q_c + q_s)^2}{2^n} + \frac{q_c q_s}{2^a} + \frac{q_s}{2^\ell},$$

where q_c is the number of construction queries to either TRP or the random function, q_s is the number of primitive queries (to either P or the simulator $\mathcal{S}$), n is the permutation size, a is the number of truncated bits, and ℓ is the length of a fixed prefix. This bound was later improved by Choi et al. [CLL19], who showed that TRP is regularly indifferentiable from a public random function up to $\min\{2^{(n+m)/3}, 2^m, 2^\ell\}$ queries.

Indifferentiability for SUMPIP. The Indifferentiability of the SUMPIP construction was studied in [DPP08], where the authors proved that the construction is indifferentiable from a public random function up to $2^{n/2}$ queries, where n is the block size of the underlying public permutation.

Indifferentiability for SoP. In contrast to TRP and SUMPIP, the Sum of Permutations (SoP) construction has been extensively analysed in the indifferentiability framework. The study began with the work of Mandal et al. [MPN10], who proved that SoP, instantiated with two independent n-bit public permutations, is indifferentiable from a public random function up to $2^{2n/3}$ queries. This result was revisited by Mennink and Preneel [MP15], who identified a flaw in the original analysis and established security up to $2^{2n/3}/n$ queries. Bhattacharya and Nandi [BN18] later proved the optimal 2^n-query indifferentiability security of SoP using a refined simulator. Further, Lee [Lee17] generalised the result, showing that the sum of r independent n-bit permutations achieves security up to $2^{(r-1)n/r}$ queries for even integers $r \geq 4$. All these results were originally claimed to hold in the regular indifferentiability framework. However, Gunsing [Gun22] observed that the existing proofs are actually valid under the sequential indifferentiability model as the analysis implicitly assumed that all primitive queries are done before any construction query. In a recent work, Gunsing et al. [GBJ+23] analysed the SoP construction in the regular indifferentiability model. They identified a flaw in the proof by Bhattacharya and Nandi [BN18], and showed that SoP achieves only a $2^{2n/3-\log n}$-query security bound in the regular indifferentiability model.

1.3 Our Contribution

In the last few years, while a long line of research has focused on analyzing the security of SoP, TRP, and SUMPIP, the indifferentiability framework has not yet been applied to other block cipher based PRFs. In this paper, we initiate the first such study by analysing the security of two popular constructions, STH and EDM, along with their single-key variants, in the sequential indifferentiability model. Following is the summary of the contributions:

1. We first demonstrate that the STH and STH2 constructions are not sequentially indifferentiable from a random function, even against a distinguisher that makes only one primitive query and one construction query. To address this limitation, we propose a generalised variant of STH, denoted by gSTH, which employs domain separation using two l-bit constants instead of the single-bit values '0' and '1'. The

Table 1: Sequential and Regular Indifferentiability Results of PRP-based PRFs. The symbols "?" and "×" mean Not known and insecure, respectively. We use the symbol (†) to denote that the bound is tight.

Construction	Sequential	Regular	Reference
TRP	$\min\{2^{(n+m)/3}, 2^m, 2^l\}$	$\min\{2^{(n+m)/3}, 2^m, 2^l\}$	[CLL19]
SUMPIP	$2^{n/2}$	?	[DPP08]
SoP	$2^{2n/3-\log n}$	$2^{2n/3-\log n}$	[GBJ+23]
STH	×	×	Sect. 3.1
STH2	×	×	Sect. 3.2
gSTH	2^l (†)	?	Sect. 3.3
EDM	$2^{n/2}$	?	Sect. 4.1
P-EDM	×	×	Sect. 4.3

gSTH construction accepts an $(n-l)$-bit input and produces an $(n+m)$-bit output. We prove that gSTH is sequentially indifferentiable from a public random function for up to 2^l many possible queries. Furthermore, we present a matching attack with the same query complexity, thereby establishing the tightness of the proven security bound. The simulator design, attack description, and advantage analysis are presented in Sect.3 and Sect.3.5.

2. As our next contribution, we show that the EDM is sequentially indifferentiable from a public random function up to $2^{n/2}$ queries. We have also demonstrated a matching attack relative to our proposed simulator, thereby demonstrating the tightness of the proven bound specific to the simulator. However, our result does not imply that the sequential indifferentiability bound for EDM is optimal; it remains open if an alternative simulator could achieve an improved bound. Full details of the simulator, the attack, and the advantage analysis are provided in Sect.4 and Sect.4.2. In addition, we show that EDM instantiated with a single permutation is not sequentially indifferentiable from a public random function. We mount a distinguishing attack against this variant that succeeds with a constant number of queries.

Table 1 summarizes the existing indifferentiability bounds of PRP-based PRF constructions, including our results.

Open Problems. We have analyzed the security of the constructions in a weaker indifferentiability model. It remains open to analyze the security of EDM and gSTH in the regular indifferentiability model. We would like to mention here that we have not analyzed the security of EDMD in terms of the indifferentiability framework. This is because EDMD can be equivalently viewed as the SoP construction. Thus, the indifferentiable bound of the SoP is carried forward to the indifferentiable security bound of EDMD. However, analyzing the indifferentiable security of single-keyed EDMD is left as a future research problem.

1.4 Why Sequential Indifferentiability Matters: A Deep Dive

As mentioned earlier, the notion of sequential indifferentiability restricts the distinguisher from making primitive queries after construction queries. Although sequential indifferentiability is weaker than regular indifferentiability, it still implies an important property known as correlation intractability [CGH98]. Informally, this means that if a construction C is sequentially indifferentiable from an ideal primitive F, then any relation that is hard to find for F remains hard to find for C^{P}, when P is chosen at random. The connection between correlation intractability and sequential indifferentiability was formally established in [MPS12, Theorem 3].

> *Let C be a construction with oracle access to an ideal primitive P and implementing some primitive F. If C^{P} is statistically (resp. computationally) sequential indifferentiable from the ideal primitive F, then C^{P} is correlation intractable (resp. weakly correlation intractable).*

Knudsen and Rijmen [KR07] proposed correlation intractability as a theoretical foundation for capturing the chosen-key attack model, which was later formalised in a rigorous framework in [CS15]. It is well known that defining security against chosen-key attacks in the standard model leads to impossibility results similar to those shown in [CGH98] regarding random oracles. To circumvent this, the formalisation was developed in idealised models. Thus, sequential indifferentiability not only guarantees correlation intractability but also provides a level of resistance against chosen-key attacks.

Additionally, sequential indifferentiability is simpler to work with and often easier to use in proofs compared to regular indifferentiability. Indeed, several constructions have been analysed under the sequential indifferentiability model, including those in [MPS12, CS15, CS16b, CEJS23, CPS08, LS13, MPN10, MP15, Lee17, BN18]. While regular indifferentiability remains the ultimate goal for security proofs, sequential indifferentiability serves as a valuable intermediate step. Gunsing et al. [GBJ+23] introduced a security lifting lemma that decomposes the regular indifferentiability advantage into three steps: (1) constructing a simulator and bounding the advantage in the sequential indifferentiability model, (2) bounding the failure probability of the simulator in the regular setting, and finally (3) bounding the sequential difference. This decomposition allows one to lift a proof of sequential indifferentiability to the regular indifferentiability by addressing components (2) and (3). It is important to note that step (3) is highly non-trivial, and it has been largely overlooked in prior works on SoP [MPN10, MP15, BN18, Lee17]. Following this approach, sequential indifferentiability can thus serve as a practical stepping stone toward achieving regular indifferentiability. This methodology was successfully applied by Gunsing et al. [GBJ+23] in their proof of the regular indifferentiability of the SoP construction. They used the simulator designed by Mennink et al. [MP15] for the sequential model as a starting point and enhanced it to satisfy the regular model. Therefore, demonstrating the sequential indifferentiability of cryptographic constructions is not only meaningful in its own right but also instrumental in building toward full indifferentiability guarantees. We believe that our results on the sequential indifferentiability of the aforementioned constructions provide a strong foundation for future work aimed at establishing their regular indifferentiability bounds.

2 Preliminaries

Notation. Let $n \in \mathbb{N}$ be a natural number. $\{0,1\}^n$ denotes the set of all n-bit binary strings, and $\{0,1\}^*$ denotes the set of all binary strings of finite length. For $n, m \geq 1$ and $n > m$, $x_1 \| x_2 \xleftarrow{m,n-m} x$ denotes that the n-bit string x is parsed into two parts: first m-bit

string x_1 followed by the remaining $n-m$ bit string x_2. We write $\mathrm{left}_m(x)$ to denote the most significant m bits of the n-bit string x and $\mathrm{right}_m(x)$ to denote the least significant m bits of the n-bit string x. For an integer i, $\langle i \rangle_s$ denotes the s-bit binary representation of the integer i. Similarly, for a binary string k, $\mathsf{int}(k)$ denotes the equivalent decimal representation of the binary string k. For two binary strings $x, y \in \{0,1\}^*$, we write $x \| y$ to denote the concatenation of x and y. For two n-bit binary strings x, y, we write $x \oplus y$ to denote the xor of x and y. We write $[a, b]$ to denote the set $\{a, a+1, \ldots, b\}$, where $a \leq b$. We denote the set of all functions that map n_1 bit strings to n_2 bit strings by $\mathsf{Func}(n_1, n_2)$. When $n_1 = n_2 = n$, we simply write it as $\mathsf{Func}(n)$. We write the set of all n-bit permutations as $\mathsf{Perm}(n)$. We write $X \xleftarrow{\$} \mathcal{X}$ to denote that the random variable X is uniformly sampled from the set $\mathcal{X}$. We call a random variable RF to be an n-bit random function if $\mathsf{RF} \leftarrow\!\$\, \mathsf{Func}(n)$ and RP to be an n-bit random permutation if $\mathsf{RP} \leftarrow\!\$\, \mathsf{Perm}(n)$. Let $\mathscr{D} = \{(x, y) : x, y \in \{0,1\}^n\}$ be the set of all n-bit pairs of binary strings. We write $\mathsf{Dom}(\mathscr{D})$ to be the set of all x values, i.e., $\mathsf{Dom}(\mathscr{D}) = \{x : (x, \star) \in \mathscr{D}\}$ and $\mathsf{Ran}(\mathscr{D})$ denotes the set of all y values, i.e., $\mathsf{Ran}(\mathscr{D}) = \{y : (\star, y) \in \mathscr{D}\}$.

Distinguishing Advantage. Let F and G be two random systems, and an adversary $\mathcal{D}$ is given oracle access to either of F or G. After interaction with an oracle $\mathcal{O} \in \{\mathsf{F}, \mathsf{G}\}$, $\mathcal{D}$ outputs 1, which is denoted as $\mathcal{D}^{\mathcal{O}} \to 1$. Such an adversary is called a *distinguisher* and the game is called a *distinguishing game*. The task of the distinguisher in a distinguishing game is to tell which of the two systems it has interacted with. The advantage of the distinguisher $\mathcal{D}$ in distinguishing the random system F from G is defined as

$$\mathbf{Adv}^{\mathsf{F}}_{\mathsf{G}}(\mathcal{D}) := \mid \Pr[\mathcal{D}^{\mathsf{F}} \to 1] - \Pr[\mathcal{D}^{\mathsf{G}} \to 1] \mid,$$

where the above probability is defined over the probability spaces of $\mathcal{D}$ and $\mathcal{O}$. The maximum advantage in distinguishing F from G is defined as $\max_{\mathcal{D} \in \mathcal{C}} \mathbf{Adv}^{\mathsf{F}}_{\mathsf{G}}(\mathcal{D})$, where $\mathcal{C}$ is the class of all possible probabilistic polynomial time distinguishers.

2.1 Indifferentiability

In the indifferentiability framework, a distinguisher is given two systems $(\mathsf{C}^{\mathsf{P}}, \mathsf{P})$ and $(\mathsf{F}, \mathcal{S}^{\mathsf{F}})$, where P is an ideal primitive, C^{P} is a bigger construction using P as a building block, F is another ideal primitive with the same interface as C^{P}, and $\mathcal{S}$ is a simulator with oracle access to F. The goal of the simulator is to emulate the ideal primitive P so that no distinguisher can tell apart the two systems $(\mathsf{C}^{\mathsf{P}}, \mathsf{P})$ and $(\mathsf{F}, \mathcal{S}^{\mathsf{F}})$ with a significant probability, based on their responses to queries that the distinguisher may send. We say that the construction C^{P} is indifferentiable from the ideal primitive F if the existence of such a simulator is proved. Here we consider two different variants of the indifferentiability notion as follows:

Definition 1 (Regular Indifferentiability). A construction C with oracle access to an ideal primitive P is said to be (q, σ, ϵ)-indifferentiable from an ideal primitive F if there exists a simulator $\mathcal{S}$ with oracle access to F such that for any distinguisher $\mathcal{D}$ making a total of q queries to the primitive and the simulator makes a total of σ queries to the ideal primitive F, it holds that

$$\mathbf{Adv}^{\text{reg-indiff}}_{\mathsf{C},\mathcal{S}}(\mathcal{D}) = \left| \Pr\left[\mathcal{D}^{\mathsf{C}^{\mathsf{P}},\mathsf{P}} \to 1\right] - \Pr\left[\mathcal{D}^{\mathsf{F},\mathcal{S}^{\mathsf{F}}} \to 1\right] \right| < \epsilon.$$

We say that C^{P} is computationally indifferentiable from the ideal primitive F if ϵ is a negligible function of the security parameter.

Regular indifferentiability guarantees universal composability of C^{P} [MRH04,CDMP05]: if a construction C^{P} is regular indifferentiable from an ideal primitive F, then any cryptosystem

proven secure when used with F in the single stage game [RSS11], remains secure when used with the construction C^{P}.

Definition 2 (**Sequential Indifferentiability).** A construction C with oracle access to an ideal primitive P is said to be sequentially (q, σ, ϵ)-indifferentiable from an ideal primitive F if there exists a simulator $\mathcal{S}$ with oracle access to F such that for any distinguisher $\mathcal{D}$ making exactly q queries to the primitive and the simulator makes a total of σ queries to the ideal primitive F such that the distinguisher is restricted in first making its primitive queries and then making its construction queries, it holds that

$$\mathbf{Adv}^{\text{seq-indiff}}_{\mathsf{C},\mathcal{S}}(\mathcal{D}) = \left|\Pr\left[\mathcal{D}^{\mathsf{C}^{\mathsf{P}},\mathsf{P}} \to 1\right] - \Pr\left[\mathcal{D}^{\mathsf{F},\mathcal{S}^{\mathsf{F}}} \to 1\right]\right| < \epsilon.$$

We say that C^{P} is computationally sequential indifferentiable from the ideal primitive F if ϵ is a negligible function of the security parameter.

2.2 H-Coefficient Technique

Let $\mathcal{D}$ be a computationally unbounded deterministic distinguisher that interacts with either the real world or the ideal world. The collection of all the queries and responses that $\mathcal{D}$ made and received to and from the oracle, is called the *transcript* of $\mathcal{D}$, denoted as τ. Sometimes, we allow the oracle to release more internal information to $\mathcal{D}$ only after $\mathcal{D}$ completes all its queries and responses, but before it outputs its decision bit. Let X_{re} denote the random variable that takes a transcript τ realized in the real world. Similarly, X_{id} denotes the random variable that takes a transcript τ realized in the ideal world. The probability of realizing a transcript τ in the ideal (resp. real) world is called *ideal* (*resp. real*) *interpolation probability*. A transcript τ is said to be attainable with respect to a distinguisher $\mathcal{D}$, if its ideal interpolation probability is non-zero. Let Θ denote the set of all attainable transcripts. Following these notations, we state the main theorem of the H-Coefficient technique as follows:

Theorem 1 (**H-Coefficient**). *Let, $\mathcal{D}$ be a fixed deterministic distinguisher that has access to either the real oracle $\mathcal{O}_{\text{re}}$ or the ideal oracle $\mathcal{O}_{\text{id}}$. Let, $\Theta = \mathsf{Good} \sqcup \mathsf{Bad}$ be some partition of the set of attainable transcripts Θ. For any good attainable transcript $\tau \in \mathsf{Good}$, let*

$$\frac{\mathsf{p}_{\text{re}}(\tau)}{\mathsf{p}_{\text{id}}(\tau)} = \frac{\Pr[\mathsf{X}_{\text{re}} = \tau]}{\Pr[\mathsf{X}_{\text{id}} = \tau]} \geq 1 - \epsilon_{\text{good}},$$

for some $\epsilon_{\text{good}} \geq 0$, and there exists $\epsilon_{\text{bad}} \geq 0$ such that $\Pr[\mathsf{X}_{\text{id}} \in \mathsf{Bad}] \leq \epsilon_{\text{bad}}$ holds. Then,

$$\mathbf{Adv}^{\mathcal{O}_{\text{id}}}_{\mathcal{O}_{\text{re}}}(\mathcal{D}) := |\Pr[\mathcal{D}^{\mathcal{O}_{\text{re}}} \to 1] - \Pr[\mathcal{D}^{\mathcal{O}_{\text{id}}} \to 1]| \leq \epsilon_{\text{good}} + \epsilon_{\text{bad}}. \tag{1}$$

3 Sequential Indifferentiability of STH, STH2 and a Generalized Version

Summation Truncation Hybrid (STH), introduced by Gunsing and Mennink [GM20], is an $n-1$-bit to $n+m$ bit PRF based on an n-bit random permutation P. For an $n-1$ bit string x, it produces the output as

$$\text{STH}(x) := \text{left}_m(\mathsf{P}(x\|0)) \parallel \text{left}_m((\mathsf{P}(x\|1)) \parallel \text{right}_{n-m}(\mathsf{P}(x\|0) \oplus \mathsf{P}(x\|1)).$$

Authors have shown that STH achieves $n - m/2$ bit indistinguishable security bound. Note that, STH is based on an n-bit random permutation P with input domain separation using bit 0 and bit 1. However, a two-keyed variant of the construction, called STH2, had also

been proposed in [GM20] with a similar indistinguishable security bound as that of STH. For an n bit string x, the construction STH2 produces the output as

$$\mathrm{STH2}(x) := \mathrm{left}_m(\mathsf{P}_1(x)) \parallel \mathrm{left}_m(\mathsf{P}_2(x)) \parallel \mathrm{right}_{n-m}(\mathsf{P}_1(x) \oplus \mathsf{P}_2(x)).$$

A pictorial description of the STH and STH2 constructions is depicted in Fig. 1 and Fig. 2, respectively.

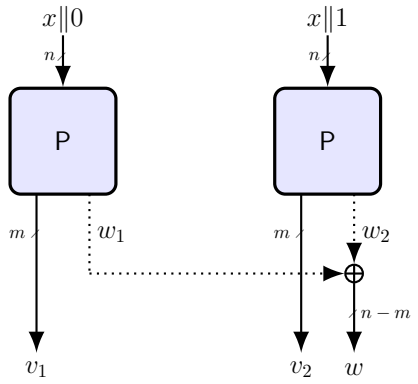

Figure 1: Summation Truncation Hybrid (STH) Construction based on an n-bit random permutation P

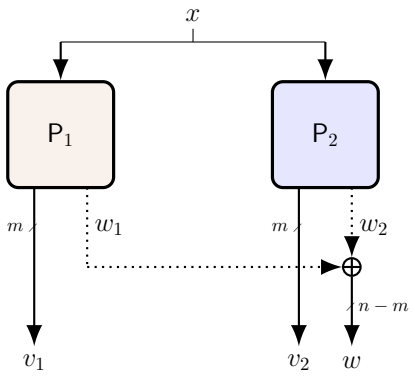

Figure 2: STH2 Construction based on two n-bit independent random permutations P_1 and P_2.

3.1 Indifferentiable Attack on STH

This section presents an attack on the STH construction in the indifferentiable setting by making only one primitive query and one construction query. This attack relies on the difficulty of finding a pre-image of an element whose m-bits (z_1 or z_2) are selected by the distinguisher for a random function, by any computationally bounded simulator. The complete attack description is given in Fig. 3.

Generic Attack Algorithm for STH

1 : Make inverse primitive query with 0^n;
2 : Let u be the response;
3 : Make construction query with $\mathrm{left}_{n-1}(u)$;
4 : Let $z_1 \| z_2 \| z_3$ be the response;
5 : **if** $((\mathrm{right}_1(u) = 0 \wedge z_1 = 0^m) \vee (\mathrm{right}_1(u) = 1 \wedge z_2 = 0^m))$
6 : **return** "1";
7 : **else**
8 : **return** "0";

Figure 3: Differentiability Attack against the STH construction.

Analysis of the Attack. Now, we calculate the success probability of the distinguisher, described in Fig. 3, against any simulator for the STH construction. To achieve this, we

recall the indifferentiability security definition

$$\mathbf{Adv}^{\text{seq-indiff}}_{\text{STH},\mathcal{S}}(\mathcal{D}) = \left|\Pr[\mathcal{D}^{\mathsf{W}_{\text{re}}} \to 1] - \Pr[\mathcal{D}^{\mathsf{W}_{\text{sim}}} \to 1]\right|, \tag{2}$$

where W_{re} is the real world oracle, i.e., STH construction and W_{sim} is the simulated world oracle.

Bounding $\Pr[\mathcal{D}^{\mathsf{W}_{\text{re}}} \to 1]$. It is easy to see that while interacting with the real world (STH, P), the distinguisher only outputs 0 if there is any inconsistency between primitive and construction queries, and hence, we have $\Pr[\mathcal{D}^{\mathsf{W}_{\text{re}}} \to 1] = 1$.

Bounding $\Pr[\mathcal{D}^{\mathsf{W}_{\text{sim}}} \to 1]$. Note that, while interacting with the simulator world, $\mathcal{D}$ returns 1 only if, the simulator can find either $u = x\|0$ such that $\mathsf{RF}(x) = 0^m\| * \|*$, or $u = x\|1$ such that $\mathsf{RF}(x) = *\|0^m\|*$. Note that, for any $x \in \{0,1\}^{n-1}$, $\Pr[\mathsf{RF}(x) = 0^m\| * \|*] = \Pr[\mathsf{RF}(x) = *\|0^m\|*] = 2^{-m}$. Since we consider polynomial time simulators, it can try at most $p(n)$ times to find such an u. Thus, the probability of finding the desired $x\|0$ or $x\|1$ is upper bounded by $\frac{2p(n)}{2^m}$, and we obtain $\Pr[\mathcal{D}^{\mathsf{W}_{\text{sim}}} \to 1] \leq \frac{2p(n)}{2^m}$, where p is a polynomial.

Combining the above two probabilities, we have

$$\mathbf{Adv}^{\text{seq-indiff}}_{\text{STH},\mathcal{S}}(\mathcal{D}) \geq \left|1 - \frac{2p(n)}{2^m}\right|. \tag{3}$$

3.2 Indifferentiable Attack on STH2

This section presents an attack on the STH2 construction in the indifferentiability setting, using only one primitive query and one construction query. The attack closely follows the procedure used for the STH construction. The complete attack description is given in Fig. 4.

Generic Attack Algorithm for STH2

```
1 : Make inverse primitive query with 0^n to P_1;
2 : Let x be the response;
3 : Make construction query with x;
4 : Let z_1||z_2||z_3 be the response;
5 : if (z_1 = 0^m)
6 :    return "1";
7 : else
8 :    return "0";
```

Figure 4: Differentiability Attack against the STH2 construction.

Analysis of the Attack. Now, we calculate the success probability of the distinguisher, described in Fig. 4, against any simulator for the STH2 construction. To achieve this, we recall the indifferentiability security definition

$$\mathbf{Adv}^{\text{seq-indiff}}_{\text{STH2},\mathcal{S}}(\mathcal{D}) = \left|\Pr[\mathcal{D}^{\mathsf{W}_{\text{re}}} \to 1] - \Pr[\mathcal{D}^{\mathsf{W}_{\text{sim}}} \to 1]\right|, \tag{4}$$

where W_{re} is the real world oracle, i.e., STH construction and W_{sim} is the simulated world oracle.

Bounding $\Pr[\mathcal{D}^{\mathsf{W}_{\mathrm{re}}} \to 1]$. It is easy to see that while interacting with the real world $(\mathrm{STH2}, \mathsf{P})$, the distinguisher only outputs 0 if there is any inconsistency between primitive and construction queries, and hence, we have $\Pr[\mathcal{D}^{\mathsf{W}_{\mathrm{re}}} \to 1] = 1$.

Bounding $\Pr[\mathcal{D}^{\mathsf{W}_{\mathrm{sim}}} \to 1]$. Note that, while interacting with the simulator world, $\mathcal{D}$ returns 1 only if, the simulator can find x such that $\mathsf{RF}(x) = 0^m \| * \| *$. Note that, for any $x \in \{0,1\}^n$, $\Pr[\mathsf{RF}(x) = 0^m \| * \| *] = 2^{-m}$. Since we consider polynomial time simulators, it can try at most $p(n)$ times to find such an x. Thus, the probability of finding the desired x is upper bounded by $\frac{p(n)}{2^m}$, and we obtain $\Pr[\mathcal{D}^{\mathsf{W}_{\mathrm{sim}}} \to 1] \leq \frac{p(n)}{2^m}$, where p is a polynomial.

Combining the above two probabilities, we have

$$\mathbf{Adv}^{\text{seq-indiff}}_{\mathrm{STH2},\mathcal{S}}(\mathcal{D}) \geq \left| 1 - \frac{p(n)}{2^m} \right|. \tag{5}$$

3.3 Generalized STH and Its Sequential Indifferentiability

This section proposes a generalized variant of the STH construction dubbed as gSTH. Fix two constants l and m, such that $l, m < n$. The gSTH construction (Fig. 5) takes an input $x \in \{0,1\}^{n-l}$ and provides an output of $\{0,1\}^{n+m}$ as follows:

$$\mathrm{gSTH}(x) = \mathrm{left}_m(\mathsf{P}(x \| c_1)) \parallel \mathrm{left}_m((\mathsf{P}(x \| c_2)) \parallel \mathrm{right}_{n-m}(\mathsf{P}(x \| c_1) \oplus \mathsf{P}(x \| c_2)),$$

where $c_1 \neq c_2 \in \{0,1\}^l$. Note that, the STH construction is a special case of gSTH construction with $l = 1$ and $c_1 = 0,\ c_2 = 1$. We prove that gSTH is indifferentiable from a random function RF with a security bound of $\mathcal{O}\left(q/2^l\right)$, as stated in Theorem 2.

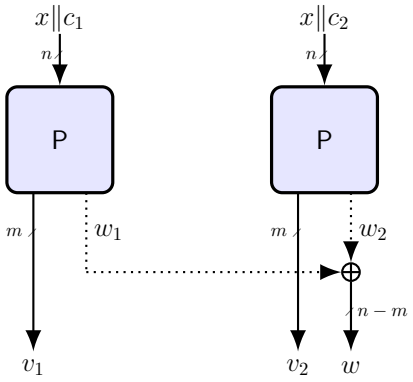

Figure 5: Generalized Summation Truncation Hybrid (gSTH) Construction based on an n-bit random permutation P

Theorem 2. *There exists a simulator $\mathcal{S}$ such that for any distinguisher $\mathcal{D}$ making exactly q_c construction queries and q_s primitive queries, one has*

$$\mathbf{Adv}^{\text{seq-indiff}}_{\mathrm{gSTH},\mathcal{S}}(\mathcal{D}) \leq \frac{2q}{2^{l-2}} + \frac{2q}{2^n},$$

where $q = q_c + q_s$ denotes the total number of queries and we assume that, $q_s \leq 2^{n-m}$ and $q < 2^{n-1}$.

3.4 Proof of Theorem 2

Let $\mathsf{W}_{\text{sim}} = (\mathsf{RF}, \mathcal{S}[\mathsf{RF}])$ and $\mathsf{W}_{\text{re}} = (\text{gSTH}[\mathsf{P}], \mathsf{P})$ denote the simulated world and the real world, respectively. We describe the simulator $\mathcal{S}$ used in our work for the gSTH[P] construction in Fig. 6. The goal of the simulator is to emulate the permutations $\mathcal{P} := (\mathsf{P}, \mathsf{P}^{-1})$ in a manner that renders $\mathcal{P}$ and $\mathcal{S}$ indistinguishable. So, $\mathcal{S}$ has interfaces corresponding to the forward and inverse queries of the random permutation P. Formally, the simulator consists of two stateful randomized algorithms $\mathcal{S}^{+}_{\mathsf{P}}$ and $\mathcal{S}^{-}_{\mathsf{P}}$, which are invoked for the responses of the forward and inverse queries of P respectively. Note that, $\mathcal{S}$ has access to the random function RF which takes input from $\{0,1\}^{n-l}$ and returns uniformly random elements of $\{0,1\}^{n+m}$. The simulator will simulate the output behavior of P and P^{-1} in the simulated world in such a way that it remains consistent with the gSTH construction, which is given by the condition

$$\mathsf{RF}(x) = \text{left}_m(\mathcal{S}^{+}_{\mathsf{P}}(x\|c_1))\|\text{left}_m(\mathcal{S}^{+}_{\mathsf{P}}(x\|c_2))\|\text{right}_{n-m}(\mathcal{S}^{+}_{\mathsf{P}}(x\|c_1) \oplus \mathcal{S}^{+}_{\mathsf{P}}(x\|c_2)),$$

for all $x \in \{0,1\}^{n-l}$. However, the simulator may fail to uphold this condition in some cases and returns $\perp$. Let $\mathcal{C} := \{v\|c_1 : v \in \{0,1\}^{n-l}\} \cup \{v\|c_2 : v \in \{0,1\}^{n-l}\}$ be the set of all n bit strings whose last l bits are either c_1 or c_2.

Internal States and Algorithm of the Simulator. The simulator $\mathcal{S}$ is designed to mimic the behavior of the primitive P while maintaining consistency with past queries and responses. It operates statefully, storing all previous interactions in the set $\mathscr{D}$ to ensure that responses are consistent and non-repetitive. The simulator handles both forward and inverse queries by strategically generating outputs that appear random to the distinguisher, except where structural dependencies are exploited as follows:.

1. **Forward Queries:** When processing a forward query, the simulator either returns a previously known response (if the query exists) or samples a new response that avoids collisions. For specific inputs with rightmost bits c_1 or c_2, the simulator uses the random function to maintain consistency with the construction's internal behavior.
2. **Inverse Queries:** For inverse queries, the simulator ensures that the pre-image of the given output is not already known, thus preserving the randomness and avoiding conflicts with earlier responses. It samples new inputs uniformly while respecting the internal state.

The key intuition is that the simulator simulates P using randomization, while leveraging the stored history to maintain coherence and avoid detection by the distinguisher. A detailed algorithmic description of the simulator is shown in Fig. 6.

Proof intuition. To bound the distinguishing advantage between the simulated world W_{sim} and the real world W_{re}, we introduce an intermediate world W_{int}. We then separately bound the distinguishing advantage between the intermediate and real worlds, and between the simulated and intermediate worlds. The former is bounded by the probability of certain bad events occurring, while the latter is addressed by first bounding the probability of inconsistent events and then applying the H-coefficient technique.

The Intermediate World W_{int}. The intermediate world is comprised of two oracles $(\mathcal{O}_{\text{cons}}, \mathcal{O}_{\text{prim}})$ that has the same oracle interface as W_{sim} and W_{re}. The oracle $\mathcal{O}_{\text{prim}}$ is built upon two algorithms, $\mathsf{P}^{(+)}_{\star}$ and $\mathsf{P}^{(-)}_{\star}$, and an internal state $\mathscr{D}_{\star}$. The intermediate world mimics a random permutation for both forward and inverse queries. The key difference is that, in the intermediate world, a $\texttt{Bad}$ flag can be triggered during inverse queries, which does not occur in the real world. An algorithmic description of $\mathsf{P}^{(+)}_{\star}$ and $\mathsf{P}^{(-)}_{\star}$ are given in the Fig. 7.

Procedure $\mathcal{S}^{+}_{\mathsf{P}}(u)$

1 : **if** $(\exists\ v\|w : (u, v\|w) \in \mathscr{D})$
2 : **return** $v\|w$;
3 : **if** $(\mathrm{right}_l(u) \neq c_1 \wedge \mathrm{right}_l(u) \neq c_2)$
4 : $v\|w \xleftarrow{\$} \{0,1\}^n \setminus \{v'\|w' : (*, v'\|w') \in \mathscr{D}\}$;
5 : $\mathscr{D} \leftarrow \mathscr{D} \cup \{(u, v\|w)\}$;
6 : **return** $v\|w$;
7 : $z_1\|z_2\|z_3 \leftarrow \mathsf{RF}(\mathrm{left}_{n-l}(u))$;
8 : **if** $(z_1 = z_2 \wedge z_3 = 0)$
9 : **return** $\perp$;
10 : $\mathcal{W} \leftarrow \{w'_1 : (*, *\|w'_1) \in \mathscr{D}\}$;
11 : $\mathcal{W}' \leftarrow \{z_3 \oplus w'_2 : (*, *\|w'_2) \in \mathscr{D}\}$;
12 : $w_1 \xleftarrow{\$} \{0,1\}^{n-m} \setminus (\mathcal{W} \cup \mathcal{W}')$;
13 : $v_1 \leftarrow z_1;\ v_2 \leftarrow z_2;\ w_2 \leftarrow z_3 \oplus w_1$;
14 : **if** $(\mathrm{right}_l(u) = c_1)$
15 : $\mathscr{D} \leftarrow \mathscr{D} \cup \{(u, v_1\|w_1)\}$;
16 : $\mathscr{D} \leftarrow \mathscr{D} \cup \{(\mathrm{right}_{n-l}(u)\|c_2, v_2\|w_2)\}$;
17 : **return** $v_1\|w_1$;
18 : **else**
19 : $\mathscr{D} \leftarrow \mathscr{D} \cup \{(u, v_2\|w_2)\}$;
20 : $\mathscr{D} \leftarrow \mathscr{D} \cup \{(\mathrm{right}_{n-l}(u)\|c_1, v_1\|w_1)\}$;
21 : **return** $v_2\|w_2$;

Procedure $\mathcal{S}^{-}_{\mathsf{P}}(v\|w)$

1 : **if** $(\exists\ u : (u, v\|w) \in \mathscr{D})$
2 : **return** u;
3 : **else**
4 : $\mathcal{U} \leftarrow \big(\{u' : (u', *) \in \mathscr{D}\} \cup \mathcal{C}\big)$;
5 : $u \xleftarrow{\$} \{0,1\}^n \setminus \mathcal{U}$;
6 : $\mathscr{D} \leftarrow \mathscr{D} \cup \{(u, v\|w)\}$;
7 : **return** u;

Figure 6: Behavior of the Simulator $\mathcal{S}$ for forward and inverse primitive queries.

Procedure $\mathsf{P}^{(+)}_{\star}(u)$

1 : **if** $(\exists\ z : z \in \mathscr{D}_\star)$
2 : **return** z;
3 : **else**
4 : $\mathcal{V} \leftarrow \{z' : (*, z') \in \mathscr{D}_\star\}$;
5 : $z \xleftarrow{\$} \{0,1\}^n \setminus \mathcal{V}$;
6 : $\mathscr{D}_\star \leftarrow \mathscr{D}_\star \cup \{(u, z)\}$;
7 : **return** z;

Procedure $\mathsf{P}^{(-)}_{\star}(z)$

1 : **if** $(\exists\ u : (u, z) \in \mathscr{D}_\star)$
2 : **return** u;
3 : **else**
4 : $\mathcal{U} \leftarrow \{u' : (u', *) \in \mathscr{D}_\star\}$;
5 : $u \xleftarrow{\$} \{0,1\}^n \setminus \mathcal{U}$;
6 : **if** $(u \in \mathcal{C})$
7 : $\mathtt{Bad} \leftarrow \mathrm{TRUE}$;
8 : $u \xleftarrow{\$} \{0,1\}^n \setminus (\mathcal{U} \cup \mathcal{C})$;
9 : $\mathscr{D}_\star \leftarrow \mathscr{D}_\star \cup \{(u, z)\}$;
10 : **return** u;

Figure 7: Algorithms of the oracle $\mathcal{O}_{\mathrm{prim}}$ for forward and inverse primitive queries.

For a construction query $x \in \{0,1\}^{n-l}$ to the oracle $\mathcal{O}_{\text{cons}}$, it invokes the primitive oracle $\mathcal{O}_{\text{prim}}$ with input $x\|c_1$ and $x\|c_2$ and obtains $v_1\|w_1 \xleftarrow{m,n-m} z_1$ and $v_2\|w_2 \xleftarrow{m,n-m} z_2$ respectively as the corresponding responses. Finally, it returns $v_1\|v_2\|(w_1 \oplus w_2)$. Let, q_c be the number of construction queries and q_s be the number of primitive queries and $q = q_c + q_s$ be the total number of queries. Then we have,

$$\begin{aligned}\mathbf{Adv}^{\text{seq-indiff}}_{\text{gSTH},\mathcal{S}}(\mathcal{D}) &= \left|\Pr[\mathcal{D}^{\mathsf{W}_{\text{sim}}} \to 1] - \Pr[\mathcal{D}^{\mathsf{W}_{\text{re}}} \to 1]\right| \\ &\leq \left|\Pr[\mathcal{D}^{\mathsf{W}_{\text{sim}}} \to 1] - \Pr[\mathcal{D}^{\mathsf{W}_{\text{int}}} \to 1]\right| \\ &\quad + \left|\Pr[\mathcal{D}^{\mathsf{W}_{\text{int}}} \to 1] - \Pr[\mathcal{D}^{\mathsf{W}_{\text{re}}} \to 1]\right|. \end{aligned} \tag{6}$$

The following two lemmas upper bound the distance between the intermediate and real world distribution and between the simulated and intermediate world distribution.

Lemma 1. *With the above definitions, one has*

$$\left|\Pr[\mathcal{D}^{\mathsf{W}_{\text{int}}} \to 1] - \Pr[\mathcal{D}^{\mathsf{W}_{\text{re}}} \to 1]\right| \leq \frac{q}{2^{l-2}}.$$

Lemma 2. *With the above definitions, one has*

$$\left|\Pr[\mathcal{D}^{\mathsf{W}_{\text{sim}}} \to 1] - \Pr[\mathcal{D}^{\mathsf{W}_{\text{int}}} \to 1]\right| \leq \frac{q}{2^{l-2}} + \frac{2q}{2^n}.$$

Proofs of the above two lemmas are postponed to Sections 5.1 and 5.2, respectively. By combining Eqn. (6), Lemma 1 and Lemma 2, the result follows. □

3.5 Generalized STH: Tightness of the Security Bound

In this subsection, we describe a distinguishing attack that essentially distinguishes between the real construction gSTH and any simulator $\mathcal{S}$. The attack essentially extends the idea used in mounting the attack against STH as follows: If a simulator returns a valid element of the form $*\|c_1$ or $*\|c_2$ for an inverse query, it must find a pre-image of an element whose m-bits (z_1 or z_2) are selected by the distinguisher for a random function, which is computationally hard for a bounded simulator. Moreover, as $|c_1| = |c_2| = l$, for distinct 2^l many inverse queries, every simulator has to return at least one valid element. An algorithmic description of the attack is presented in Fig. 8.

Generic Attack Algorithm for gSTH

```
for i = 1, ..., 2^l
  a_i ← ⟨i⟩_n;
  Make an inverse primitive queries with a_i;
  Let response be u_i;
  if ((u_i = x_i‖c_1) ∨ (u_i = x_i‖c_2))
    Make a construction query with x_i;
    Let the response be z_1‖z_2‖z_3;
    if ((u_i = x_i‖c_1 ∧ z_1 = left_m(a_i)) ∨ (u_i = x_i‖c_2 ∧ z_2 = left_m(a_i)))
      return 1;
    else
      return 0;
return 0;
```

Figure 8: Attack algorithm for the gSTH construction.

Lemma 3. *With the definition of the distinguisher $\mathcal{D}$ above, one has*

$$\mathbf{Adv}^{\text{seq-indiff}}_{\text{gSTH},\mathcal{S}}(\mathcal{D}) \geq \left|1 - \frac{2p(n)}{2^m} - \frac{1}{e}\right|.$$

Proof of the lemma is similar to the analysis carried out in Section 3.1. However, for the sake of completeness, we keep the proof in Section 5.3.

Note that, according to Theorem 2, if $q = 2^l$, the advantage is upper bounded by $O(1) + O(1/2^{n-l})$. On the other hand, from the attack perspective, if $q = 2^l$, then the advantage is lower bounded by $1 - O(1/2^{m-l})$. Since $m < n$, the advantage is upper bounded by $O(1)+O(1/2^{m-l})$, which matches with the lower bound of the attack advantage and hence the bound is tight.

4 Sequential Indifferentiability of Encrypted Davies Meyer and Its Single Permutation Variant

Encrypted Davies-Meyer (EDM), proposed by Cogliati and Seurin [CS16a], is an n-bit to n-bit PRF based on two independent n-bit permutations that achieves indistinguishability security up to $2^{2n/3}$ queries

$$\text{EDM}(x) := \mathsf{P}_2(\mathsf{P}_1(x) \oplus x)).$$

Later in [DHT17], the bound was improved from $2n/3$ bits to $3n/4$ bits. In this section, we prove that EDM is indifferentiable from an n-bit random function RF with a security bound of $\mathcal{O}(q^2/2^n)$, as stated in Theorem 3.

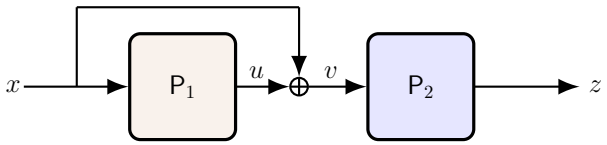

Figure 9: Encrypted Davies-Meyer Construction based on two n-bit independent random permutations P_1 and P_2.

4.1 Birthday-bound Sequential Indifferentiability of EDM

Theorem 3. *There exists a simulator $\mathcal{S}$ such that for any distinguisher $\mathcal{D}$ making exactly q_c construction queries and q_s primitive queries, one has*

$$\mathbf{Adv}^{\text{seq-indiff}}_{\text{EDM},\mathcal{S}}(\mathcal{D}) \leq \frac{4q^2}{2^n},$$

where $q = 2q_c + q_s$ denotes the total number of queries and we assume $q < 2^{n-1}$.

Proof. Let $\mathsf{W}_{\text{sim}} = (\mathsf{RF}, \mathcal{S}[\mathsf{RF}])$ and $\mathsf{W}_{\text{re}} = (\text{EDM}[\mathsf{P}_1, \mathsf{P}_2], \mathsf{P}_1, \mathsf{P}_2)$ denote the simulated world and the real world, respectively. We will describe the simulator $\mathcal{S}$ used in our work for the EDM$[\mathsf{P}_1, \mathsf{P}_2]$ construction. The goal of the simulator is to emulate the permutations $\mathsf{P} := (\mathsf{P}_1, \mathsf{P}_2, \mathsf{P}_1^{-1}, \mathsf{P}_2^{-1})$ in a manner that renders P and $\mathcal{S}$ indistinguishable. So, $\mathcal{S}$ has interfaces corresponding to the forward and inverse queries of the random permutations P_1 and P_2. Formally, the simulator consists of four stateful randomized algorithms $\mathcal{S}^{\mathsf{P}_1(+)}$, $\mathcal{S}^{\mathsf{P}_1(-)}$, $\mathcal{S}^{\mathsf{P}_2(+)}$ and $\mathcal{S}^{\mathsf{P}_2(-)}$, which are invoked for the responses of the forward and inverse queries of P_1, and P_2 respectively. Note that, $\mathcal{S}$ has access to the random function RF,

which returns random elements from $\{0,1\}^n$ on every fresh query. The simulator will simulate the output behavior of P_1, P_2, P_1^{-1} and P_2^{-1} in the simulated world in such a way that it remains consistent with the EDM construction, which is given by the condition

$$\mathsf{RF}(x) = \mathcal{S}^{\mathsf{P}_2(+)}\left(\mathcal{S}^{\mathsf{P}_1(+)}(x) \oplus x\right), \ \forall x \in \{0,1\}^n.$$

However, the simulator may fail to uphold this condition in some cases. In such cases, it returns $\perp$.

Internal States and Algorithm of the Simulator. The simulator $\mathcal{S}$ is designed to mimic the behavior of the permutations P_1 and P_2 in the simulated world, making them indistinguishable from random permutations. It uses the algorithms $\mathcal{S}^{\mathsf{P}_1(+)}$, $\mathcal{S}^{\mathsf{P}_1(-)}$, $\mathcal{S}^{\mathsf{P}_2(+)}$, and $\mathcal{S}^{\mathsf{P}_2(-)}$ for this purpose. The simulator maintains internal states $\mathscr{D}_1$ and $\mathscr{D}_2$ to ensure consistency across forward and inverse queries. For each query, it either returns a fresh random response while preserving the EDM construction or reuses a previous response if available. If a conflict occurs, such as a collision in random responses, the simulator aborts to maintain the construction's integrity. The algorithm for $\mathcal{S}^{\mathsf{P}_1(+)}$ and $\mathcal{S}^{\mathsf{P}_1(-)}$ is shown in Fig. 10, and the algorithms for $\mathcal{S}^{\mathsf{P}_2(+)}$ and $\mathcal{S}^{\mathsf{P}_2(-)}$ are shown in Fig. 11.

Procedure $\mathcal{S}^{\mathsf{P}_1(+)}(x)$

1 : **if** $(\exists\, u, s :\ (x,u,s) \in \mathscr{D}_1)$
2 : **if** $(s = 0)$
3 : $s \leftarrow \{+\}$;
4 : **return** u;
5 : $z \leftarrow \mathsf{RF}(x)$;
6 : **if** $((*,z,*) \in \mathscr{D}_2)$
7 : **return** $\perp$;
8 : **else**
9 : $\mathcal{U}_1 \leftarrow \{u' : (*,u',*) \in \mathscr{D}_1\}$;
10 : $\mathcal{U}_2 \leftarrow \{x \oplus v' : (v',*,*) \in \mathscr{D}_2\}$;
11 : $u \xleftarrow{\$} \{0,1\}^n \setminus (\mathcal{U}_1 \cup \mathcal{U}_2)$;
12 : $v \leftarrow u \oplus x$;
13 : $\mathscr{D}_1 \leftarrow \mathscr{D}_1 \cup \{(x,u,+)\}$;
14 : $\mathscr{D}_2 \leftarrow \mathscr{D}_2 \cup \{(v,z,0)\}$;
15 : **return** u;

Procedure $\mathcal{S}^{\mathsf{P}_1(-)}(u)$

1 : **if** $(\exists\, x, s :\ (x,u,s) \in \mathscr{D}_1)$
2 : **if** $(s = 0)$
3 : $s \leftarrow \{-\}$;
4 : **return** x;
5 : $\mathcal{X}_1 \leftarrow \{x' : (x',*,*) \in \mathscr{D}_1\}$;
6 : $\mathcal{X}_2 \leftarrow \{u \oplus v' : (v',*,*) \in \mathscr{D}_2\}$;
7 : $x \xleftarrow{\$} \{0,1\}^n \setminus (\mathcal{X}_1 \cup \mathcal{X}_2)$;
8 : $z \leftarrow \mathsf{RF}(x)$;
9 : **if** $((*,z,*) \in \mathscr{D}_2)$
10 : **return** $\perp$;
11 : **else**
12 : $v \leftarrow u \oplus x$;
13 : $\mathscr{D}_1 \leftarrow \mathscr{D}_1 \cup \{(x,u,-)\}$;
14 : $\mathscr{D}_2 \leftarrow \mathscr{D}_2 \cup \{(v,z,0)\}$;
15 : **return** x.

Figure 10: Behavior of the Simulator $\mathcal{S}$ for forward and inverse queries of P_1. $s = +/-$ denotes that the corresponding primitive query is in the forward/backward direction, respectively. $s = 0$ indicates that the corresponding input-output pair is determined while answering another query.

Proof Intuition. To bound the distinguishing advantage between the simulated world W_{sim} and the real world W_{re}, we begin by analyzing the behavior of an arbitrary distinguisher. We then replace construction queries of the distinguisher with primitive queries, aided by the release of some additional information. Finally, we bound the distinguishing advantage by first quantifying the probability of inconsistent events and then applying the H-coefficient technique.

Procedure $\mathcal{S}^{\mathsf{P}_2(+)}(v)$

1 : **if** $(\exists\ (v, z, s) \in \mathscr{D}_2)$
2 : **if** $(s = 0)$
3 : $s \leftarrow \{+\}$;
4 : **return** z;
5 : $\mathcal{X}_1 \leftarrow \{x' : (x', *, *) \in \mathscr{D}_1\}$;
6 : $\mathcal{X}_2 \leftarrow \{v \oplus u' : (*, u', *) \in \mathscr{D}_1\}$;
7 : $x \xleftarrow{\$} \{0,1\}^n \setminus (\mathcal{X}_1 \cup \mathcal{X}_2)$;
8 : $z \leftarrow \mathsf{RF}(x)$;
9 : **if** $((*, z, *) \in \mathscr{D}_2)$
10 : **return** $\perp$;
11 : **else**
12 : $u \leftarrow v \oplus x$;
13 : $\mathscr{D}_1 \leftarrow \mathscr{D}_1 \cup \{(x, u, 0)\}$;
14 : $\mathscr{D}_2 \leftarrow \mathscr{D}_2 \cup \{(v, z, +)\}$;
15 : **return** z;

Procedure $\mathcal{S}^{\mathsf{P}_2(-)}(z)$

1 : **if** $(\exists\ v\ :\ (v, z, s) \in \mathscr{D}_2)$
2 : **if** $(s = 0)$
3 : $s \leftarrow \{-\}$;
4 : **return** v;
5 : $\mathcal{V}_1 \leftarrow \{v' : (v', *, *) \in \mathscr{D}_2\}$;
6 : $v \xleftarrow{\$} \{0,1\}^n \setminus \mathcal{V}_1$;
7 : $\mathscr{D}_2 \leftarrow \mathscr{D}_2 \cup \{(v, z, -)\}$;
8 : **return** v;

Figure 11: Behavior of the Simulator $\mathcal{S}$ for forward and inverse queries of P_2.

Behavior of the Distinguisher $\mathcal{D}$. The distinguisher $\mathcal{D}$ can make the following two types of queries:

1. It makes both forward and inverse primitive queries, each corresponding to P_1 and P_2. Responses to these queries are provided by the simulator utilizing its designated algorithms in the simulated world, whereas in the real world, the responses are provided by invoking the actual permutations P_1 and P_2.

2. $\mathcal{D}$ makes construction queries with input $x \in \{0,1\}^n$. The real-world response to such a query is EDM(x), whereas in the simulator world, it is $\mathsf{RF}(x)$.

Moreover, due to the sequential indifferentiability setting, the distinguisher is refrained from making any primitive queries after the first construction query. Note that the simulator might not be able to answer certain queries. In that case, the game is aborted. To capture this phenomenon, we introduce an additional symbol $b' \in \{\top, \perp\}$, where $b' = \perp$ denotes that the query-response game has been aborted, whereas $b' = \top$ denotes that the query-response game has not been aborted. Note that, the game in the real world never aborts. After the interaction of the distinguisher is over, the collection of the query-response is summarized in the following transcript:

$$\tau' = (\tau'_c, \tau'_{\mathsf{P}_1}, \tau'_{\mathsf{P}_2}, b'),$$

where

- τ'_c contains all query response pairs to the construction oracle in the form of (x^i_c, z^i_c), where x^i_c is the i-th construction query and z^i_c is the corresponding response. Note that, z^i_c can either be EDM(x^i_c) or it can be $\mathsf{RF}(x^i_c)$. Therefore, $\tau'_c = \left\{(x^i_c, z^i_c)\right\}_{i=1}^{q_c}$, where q_c denotes the total number of construction queries made by the distinguisher.

- τ'_{P_1} contains all query response pairs to the primitive oracle P_1 in the form of $(x^i_{\mathsf{P}_1}, u^i_{\mathsf{P}_1}, s^i)$. Note that, these responses can either be $u^i_{\mathsf{P}_1} = \mathsf{P}_1(x^i_{\mathsf{P}_1})$ or $x^i_{\mathsf{P}_1} = \mathsf{P}_1^-(u^i_{\mathsf{P}_1})$ in the real world or $u^i_c = \mathcal{S}^{\mathsf{P}_1(+)}(x^i_{\mathsf{P}_1})$ or $x^i_{\mathsf{P}_1} = \mathcal{S}^{\mathsf{P}_1(-)}(u^i_{\mathsf{P}_1})$ in the simulated world. Therefore, $\tau_{\mathsf{P}_1} = \left\{(x^i_{\mathsf{P}_1}, u^i_{\mathsf{P}_1}, s^i)\right\}_{i=1}^{q_{p_1}}$, where q_{p_1} represents the total number of P_1 queries made by the distinguisher, and $s^i = +$ denotes a forward query, while $s^i = -$ denotes an inverse query.

- τ'_{P_2} contains all query response pairs to the primitive oracle P_2 in the form of $(v^i_{\mathsf{P}_2}, z^i_{\mathsf{P}_2}, s^i)$. Note that, these responses can either be $z^i_{\mathsf{P}_2} = \mathsf{P}_2(v^i_{\mathsf{P}_2})$ or $v^i_{\mathsf{P}_2} = \mathsf{P}_2^{-1}(z^i_{\mathsf{P}_2})$ in the real world or $z^i_{\mathsf{P}_2} = \mathcal{S}^{\mathsf{P}_2(+)}(v^i_{\mathsf{P}_2})$ or $v^i_{\mathsf{P}_2} = \mathcal{S}^{\mathsf{P}_2(-)}(z^i_{\mathsf{P}_2})$ in simulated world. Therefore, $\tau^{\mathsf{P}_2} = \left\{(v^i_{\mathsf{P}_2}, z^i_{\mathsf{P}_2}, s^i)\right\}_{i=1}^{q_{p_2}}$, where q_{p_2} represents the total number of P_2 queries made by the distinguisher, and $s^i = +$ denotes a forward query, while $s^i = -$ denotes an inverse query.

Additional Information Released to the Distinguisher. After the query-response phase is over, but before outputting the decision bit, the distinguisher is provided with the following additional information. We would like to mention that the additional information does not degrade the advantage of the adversary as it is always possible to discard it.

For each construction query (x_c, z_c), the challenger will execute the following steps:

- Make a forward primitive query to the primitive oracle P_1^+ (resp. $\mathcal{S}^{\mathsf{P}_1(+)}$) in the real world (resp. simulated world) with input x_c and obtain the response u_c.

- Perform another forward primitive query to the primitive oracle P_2^+ (resp. $\mathcal{S}^{\mathsf{P}_2(+)}$) in the real world (resp. simulated world) with input $x_c \oplus u_c$ and obtain the response z'_c.

Therefore, the distinguisher $\mathcal{D}$ receives two extra pieces of information for each construction query (x_c, z_c): (x_c, u_c) and $(x_c \oplus u_c, z'_c)$ and update the sets $\mathscr{D}_1$ and $\mathscr{D}_2$ accordingly, i.e., $\mathscr{D}_1 \leftarrow \mathscr{D}_1 \cup (x_c, u_c, +)$ and $\mathscr{D}_2 \leftarrow \mathscr{D}_2 \cup (x_c \oplus u_c, z'_c, +)$. Note that, these extra pieces of information will only increase the success probability of the distinguisher. Let

$$\tau''_{\mathsf{P}_1} \leftarrow \tau_{\mathsf{P}_1} \cup \{(x^i_c, u^i_c)\}_{i=1}^{q_c}, \tau''_{\mathsf{P}_2} \leftarrow \tau_{\mathsf{P}_2} \cup \{(x^i_c \oplus u^i_c, z'^i_c)\}_{i=1}^{q_c}.$$

Bounding Distinguishing Advantage Between $\mathbf{W}_{\text{sim}}$ and $\mathbf{W}_{\text{re}}$. Let the modified transcript equipped with additional information be denoted as $\tau'' = (\tau'_c, \tau''_{\mathsf{P}_1}, \tau''_{\mathsf{P}_2}, b'')$, where b'' is set to $\perp$ if the game is aborted (before or after releasing additional information). It is easy to see that $\tau'_{\mathsf{P}_1} \subseteq \tau''_{\mathsf{P}_1}$ and $\tau'_{\mathsf{P}_2} \subseteq \tau''_{\mathsf{P}_2}$. Note that $b' = \perp$ implies $b'' = \perp$, however, the converse may not be true. Let $|\tau''_{\mathsf{P}_1}| := q_1 \leq q_{p_1} + q_c$ and $|\tau''_{\mathsf{P}_2}| := q_2 \leq q_{p_2} + q_c$. In the following, we define two events as follows:

1. $\mathsf{E}_1 : b'' = \perp$ in τ''
2. $\mathsf{E}_2 : \exists \left(x^j_{p_1}, u^j_{p_1}, *\right) \in \tau''_{\mathsf{P}_1}, \left(v^i_{\mathsf{P}_2}, z^i_{\mathsf{P}_2}, *\right) \in \tau''_{\mathsf{P}_2}, \left(x^k_c, z^k_c\right) \in \tau'_c$: $x^j_{\mathsf{P}_1} \oplus u^j_{\mathsf{P}_1} = v^i_{\mathsf{P}_2} \wedge x^j_{\mathsf{P}_1} = x^k_c \wedge z^i_{\mathsf{P}_2} \neq z^k_c$

Let E be the union of the events E_1 and E_2. Note that, the oracle in the real world never aborts on any query, which implies that the probability of the event E_1 holds in the real world is zero. Similarly, E_2 represents an inconsistency between primitive and construction queries, which is impossible in the real world. Therefore, if either E_1 or E_2 occurs, the

distinguisher identifies the simulator with a probability of 1. Therefore,

$$\begin{aligned}\mathbf{Adv}^{\text{seq-indiff}}_{\text{EDM},\mathcal{S}}(\mathcal{D}) &= |\Pr[\mathcal{D}^{\mathsf{W}_{\text{sim}}} \to 1] - \Pr[\mathcal{D}^{\mathsf{W}_{\text{re}}} \to 1]| \\ &\leq \underbrace{|\Pr[\mathcal{D}^{\mathsf{W}_{\text{sim}}} \to 1 \mid \overline{\mathsf{E}}] - \Pr[\mathcal{D}^{\mathsf{W}_{\text{re}}} \to 1 \mid \overline{\mathsf{E}}]|}_{(\mathsf{A})} + \Pr[\mathsf{E}] \qquad (7)\end{aligned}$$

In the following, we state two lemmas which bound (A) and the probability of the event E.

Lemma 4. *With the above definitions, one has*

$$\Pr[\mathsf{E}] \leq \frac{3q^2}{2^n}.$$

Lemma 5. *With the above definitions, one has*

$$|\Pr[\mathcal{D}^{\mathsf{W}_{\text{sim}}} \to 1 \mid \overline{\mathsf{E}}] - \Pr[\mathcal{D}^{\mathsf{W}_{\text{re}}} \to 1 \mid \overline{\mathsf{E}}]| \leq \frac{q^2}{2^n}.$$

Proof of the lemmas are postponed to Sections 5.4 and 5.5, respectively. Finally, by combining Eqn. (7), Lemma 4, and Lemma 5, the result follows. □

4.2 EDM: Tightness of the Security Bound

In this subsection, we describe a distinguishing attack that essentially distinguishes between the real construction EDM and our proposed simulator $\mathcal{S}$. The attack rationale is the following: we make $2^{n/2}$ distinct forward primitive queries and $2^{n/2}$ distinct construction queries in such a way that the responses set between the primitive queries and the construction queries are disjoint with each other. The notable fact is that such a condition holds with probability 1 for the real construction, but it holds with negligible probability in the simulated world.

To facilitate this, we choose an arbitrary n-bit string u' and make distinct inverse primitive queries to P_1 with $u' \oplus i$, which we denote as u'_i, for $i = 0, 1, \ldots, 2^{n/2} - 1$. Let the response be x'_i. Then we choose another n-bit string v' in such a way so that

$$v'_i = v' \oplus i \notin \{x'_j \oplus u'_j : j = 0, 1, \ldots, 2^{n/2} - 1\},\ \forall i \in [1, 2^{n/2} - 1]. \qquad (8)$$

Then, we make forward primitive queries to P_2 with v'_i, for $i \in [1, 2^{n/2} - 1]$ and let the response be denoted as z'_i. Then we compute $2^{n/2}$ distinct n-bit binary strings $x_j = u' \oplus v' \oplus j$, for $j \in [0, 2^{n/2} - 1]$ and make construction queries with x_j to obtain the response z_j. Now, we claim that, in the real world (i.e., EDM),

$$z_j \neq z'_i,\ \forall (i, j) \in [1, 2^{n/2} - 1] \times [0, 2^{n/2} - 1].$$

For the sake of contradiction, let us assume that $z_j = z'_i$, for some i, j. It implies that $x_j \oplus \mathsf{P}_1(x_j) = v'_i$. Therefore, we have

$$\mathsf{P}_1(x_j) \ = \ x_j \oplus v'_i \ = \ u' \oplus v' \oplus j \oplus v' \oplus i \ = \ u' \oplus i \oplus j \ = \ u'_{i \oplus j} \ = \ \mathsf{P}_1(x'_{i \oplus j}).$$

This essentially says that $x_j = x'_{i \oplus j}$, and we have $v'_i = x'_{i \oplus j} \oplus u'_{i \oplus j}$ which is a contradiction with Eqn. (8). The algorithmic description of the attack is presented in Fig. 12. A formal analysis of the attack advantage is presented below.

Formal Analysis of the Attack Advantage: We begin by ensuring that if we select v' from either step 8, or step 10 of the algorithm shown in Fig. 12, then

$$v'_i \neq x'_j \oplus u'_j,\ \forall i \in [1, 2^{n/2} - 1],\ j \in [0, 2^{n/2} - 1].$$

We prove this in the following two cases:

ATTACK ON SIMULATOR FOR EDM

1 : Choose $u' \leftarrow \{0,1\}^n$; Initialize $\mathcal{L}_1 = \mathcal{L}_2 = \emptyset$;
2 : **for** $i = 0, \ldots, 2^{n/2} - 1$
3 : Compute $u'_i = u' \oplus i$;
4 : Make queries to P_1^- with u'_i; Let responses are x'_i.
5 : Compute $\mathcal{G}_i = \{x'_i \oplus u'_i \oplus 1, \ldots, x'_i \oplus u'_i \oplus (2^{n/2} - 1)\}$;
6 : Define $\mathcal{G} \leftarrow \mathcal{G}_0 \cup \mathcal{G}_1 \cup \cdots \cup \mathcal{G}_{2^{n/2}}$;
7 : **if** $(|\mathcal{G}| = 2^n)$
8 : Compute $v' = x'_1 \oplus u'_1$;
9 : **else**
10 : Choose $v' \leftarrow \{0,1\}^n \setminus \mathcal{G}$;
11 : **for** $i = 1, \ldots, 2^{n/2} - 1$
12 : Compute $v'_i \leftarrow v' \oplus i$;
13 : Make queries to P_2^+ with v'_i; Let the response be z'_i;
14 : $\mathcal{L}_1 = \mathcal{L}_1 \cup z'_i$;
15 : **for** $i = 0, \ldots, 2^{n/2} - 1$
16 : Compute $x_i = u' \oplus v' \oplus i$;
17 : Make construction queries with x_i; Let the response be z_i;
18 : $\mathcal{L}_2 = \mathcal{L}_2 \cup z_i$;
19 : **if** $(\mathcal{L}_1 \cap \mathcal{L}_2 \neq \emptyset)$
20 : **return** 0;
21 : **else**
22 : **return** 1;

Figure 12: Attack algorithm for the EDM construction

- **Case 1: v' chosen in step 8.** In this case, we have $|\mathcal{G}| = |\bigcup_{i=0}^{2^{n/2}-1} \mathcal{G}_i| = 2^n$. Hence, $|\mathcal{G}_i| = 2^{n/2}$, for all $i \in [0, 2^{n/2} - 1]$, and this implies

$$\mathcal{G}_i \cap \mathcal{G}_j = \emptyset, \ \forall i, j.$$

  Now let us assume $v'_i = x'_j \oplus u'_j$, for some i and j. Now by definition of v'_i, we have $x'_1 \oplus u'_1 \oplus i = x'_j \oplus u'_j$, and that essentially means $x'_1 \oplus u'_1 \in \mathcal{G}_j$. Thus, $\mathcal{G}_1 \cap \mathcal{G}_j \neq \emptyset$, which is a contradiction. Thus, our assumption is wrong, and $v'_i \neq x'_j \oplus u'_j$.

- **Case 2: v' chosen in step 10.** In this case, v' is chosen uniformly from the set $\{0,1\}^n \setminus \mathcal{G}$. This essentially says that $v' \neq x'_j \oplus u'_j \oplus i$, for all i and j. Hence, $v'_i \neq x'_j \oplus u'_j$, for all i and j.

Combining the above two cases, we conclude that for all i, j, $v'_i \neq x'_j \oplus u'_j$, and hence, $z_j \neq z'_i$. Hence, we obtain

$$\Pr\left[\mathcal{D}^{\mathsf{W}_{\mathrm{re}}} \rightarrow 1\right] = 1. \tag{9}$$

Let us compute the probability that the distinguisher outputs 1 while interacting with $(\mathsf{RF}, \mathcal{S})$. The simulator randomly chooses x'_i values for fresh u'_i values without replacement. The distinguisher then chooses v'_i after receiving all the (x'_i, u'_i) pairs. Next, the simulator

chooses fresh x values for each P_2^+ query with fresh v_i' and outputs $\mathsf{RF}(x)$ as z_i' values corresponding to v_i'. Note that these choices of x values are independent of the x_i values. Consequently, the values of z_i' are chosen independently from z_i values. Thus, for any (i,j) pair, $\Pr[z_i' = z_j] = 1/2^n$. Moreover, since there are $2^{n/2}$ possible options for z_i''s and $2^{n/2}-1$ possible options for z_j's, this implies that

$$\Pr[\mathcal{L}_1 \cap \mathcal{L}_2 \neq \emptyset] \;=\; \Pr[\exists\; i,j :\; z_i = z_j'] \;=\; \frac{2^{n/2}(2^{n/2}-1)}{2^n} = 1 - \frac{1}{2^{n/2}}. \tag{10}$$

So,

$$\Pr\left[\mathcal{D}^{\mathsf{W}_{\text{sim}}} \to 1\right] = \frac{1}{2^{n/2}} \tag{11}$$

Combining the definition of sequential indifferentiability, Defn. (2) with Eqn. (9) and Eqn. (11), we have

$$\begin{aligned} \mathbf{Adv}^{\text{seq-indiff}}_{\text{EDM},\mathcal{S}}(\mathcal{D}) &= \left|\Pr\left[\mathcal{D}^{\mathsf{W}_{\text{re}}} \to 1\right] - \Pr\left[\mathcal{D}^{\mathsf{W}_{\text{sim}}} \to 1\right]\right| \\ &= \left|1 - \frac{1}{2^{n/2}}\right| \end{aligned} \tag{12}$$

Remark 1. We would like to clarify that the above attack establishes the tightness of the indifferentiable bound of the EDM construction is specific to our proposed simulator. Our result does not say that EDM is indifferentiable up to $2^{n/2}$ queries. There is always room for improving the indifferentiable security bound of the construction with the help of a new simulator.

4.3 Sequential Indifferentiability of EDM with Single Permutation

Now we investigate the indifferentiability of EDM if the underlying permutations are same, i.e., $\mathsf{P}_1 = \mathsf{P}_2 = \mathsf{P}$. Let us denote this construction by P-EDM, which is depicted in Fig. 13.

We will present an attack on the P-EDM construction in the sequential indifferentiable

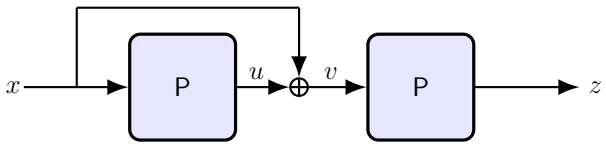

Figure 13: Encrypted Davies-Meyer Construction based on one n-bit independent random permutation P (P-EDM)

setting by making only one primitive query and one construction query. The distinguisher $\mathcal{D}$ makes an inverse primitive query with an n-bit binary string, say 0^n and let x be the response. Then $\mathcal{D}$ makes a construction query with the most significant x. Let z be the response. If $z = 0$, then it outputs 1, else output 0. The complete attack description is given in Fig. 14.

Analysis of the Attack. Now, we calculate the success probability of the distinguisher, described in Fig. 14, against any simulator for the P-EDM construction. To achieve this, we recall the indifferentiability security definition

$$\mathbf{Adv}^{\text{seq-indiff}}_{\text{P-EDM},\mathcal{S}}(\mathcal{D}) = \left|\Pr[\mathcal{D}^{\mathsf{W}_{\text{re}}} \to 1] - \Pr[\mathcal{D}^{\mathsf{W}_{\text{sim}}} \to 1]\right|, \tag{13}$$

where W_{re} is the real world oracle, i.e., (P-EDM, P) and W_{sim} is the simulated world oracle i.e., (RF, $\mathcal{S}$).

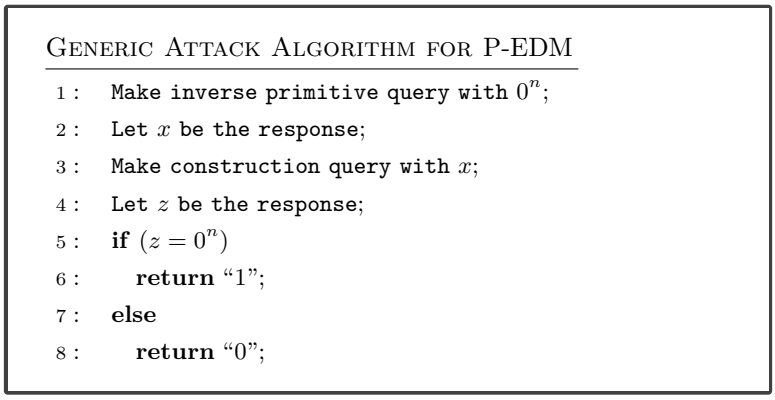

Figure 14: Distinguishing Attack against the P-EDM construction.

Bounding $\Pr[\mathcal{D}^{\mathsf{W}_{\mathrm{re}}} \to 1]$. While interacting with the real world (P-EDM, P), steps 1 and 2 of Fig. 14 confirm that $\mathsf{P}(x) = 0^n$. Then, in step 4, when the construction query is made with x, the distinguisher will get

$$\mathsf{P}(\mathsf{P}(x) \oplus x) = \mathsf{P}(0^n \oplus x) = \mathsf{P}(x) = 0^n.$$

So, the distinguisher will always output 1 and we have $\Pr[\mathcal{D}^{\mathsf{W}_{\mathrm{re}}} \to 1] = 1$.

Bounding $\Pr[\mathcal{D}^{\mathsf{W}_{\mathrm{sim}}} \to 1]$. Note that, while interacting with the simulator world, $\mathcal{D}$ returns 1 only if, the simulator can find x such that $\mathsf{RF}(x) = 0^n$. Note that, for any $x \in \{0,1\}^n$, $\Pr[\mathsf{RF}(x) = 0^n] = 2^{-n}$. Since we consider a polynomial time simulator, it can try at most $p(n)$ times to find such an x, where $p(\cdot)$ is a polynomial. Thus, the probability of finding the desired x is upper bounded by $p(n)/2^n$. Thus, we obtain $\Pr[\mathcal{D}^{\mathsf{W}_{\mathrm{sim}}} \to 1] \leq p(n)/2^n$.

Combining the above two probabilities, we have

$$\mathbf{Adv}^{\text{seq-indiff}}_{\text{P-EDM},\mathcal{S}}(\mathcal{D}) \geq \left|1 - \frac{p(n)}{2^n}\right|. \tag{14}$$

5 Auxiliary Proofs

In this section, we provide detailed formal proofs for each of the lemmas stated in Sect. 3 and Sect. 4.

5.1 Proof of Lemma 1

The algorithms of the intermediate world behave exactly like a gSTH with truly random permutation without the Bad flag being set to TRUE. Therefore,

$$\left|\Pr[\mathcal{D}^{\mathsf{W}_{\mathrm{int}}} \to 1] - \Pr[\mathcal{D}^{\mathsf{W}_{\mathrm{re}}} \to 1]\right|$$

can be upper bounded by the probability that Bad is set to true. For $i = 1, \ldots, q_s$, let E_i be the event that the i-th primitive query set Bad to true. Now since, $|\mathcal{C}| = |\mathcal{C}_1| + |\mathcal{C}_2| = 2^{n-l} + 2^{n-l} = 2^{n-l+1}$ and $|\mathscr{D}_\star| \leq q \leq 2^{n-1}$, we have

$$\Pr[\mathsf{E}_i] = \frac{|\mathcal{C}|}{2^n - (i-1)} \leq \frac{|\mathcal{C}|}{2^n - |\mathscr{D}_\star|} \leq \frac{2^{n-l+1}}{2^{n-1}} = \frac{1}{2^{l-2}} \tag{15}$$

Therefore, we have

$$\left|\Pr[\mathcal{D}^{\mathsf{W}_{\text{int}}} \to 1] - \Pr[\mathcal{D}^{\mathsf{W}_{\text{re}}} \to 1]\right| \leq \Pr\left[\bigcup_{i=1}^{q_s} \mathsf{E}_i\right] \leq \sum_{i=1}^{q_s} \Pr[\mathsf{E}_i] = \frac{q_s}{2^{l-2}} \leq \frac{q}{2^{l-2}}.$$

5.2 Proof of Lemma 2

For any distinguisher $\mathcal{D}$, there are two types of queries: (i) forward and inverse primitive queries, and (ii) construction queries. Due to the sequential indifferentiability model, the distinguisher is refrained from making any primitive queries after construction queries. After the interaction of the distinguisher is over, the collection of the query-response is summarized in the following transcript:

$$\tau' = (\tau'_c, \tau'_{\mathsf{P}}, b'),$$

where $b' \in \{\top, \bot\}$, and

- τ'_c contains all query response pairs to construction oracle in the form (x^i_c, z^i_c), where x^i_c is the i-th construction query and z^i_c is the corresponding response. Therefore, $\tau'_c = \{(x^i_c, z^i_c)\}_{i=1}^{q_c}$, where q_c denotes the total number of construction queries. We call τ'_c the *construction query transcript.*
- τ'_{P} contains all query response pairs to the primitive oracle in the form $(u^i, v^i\|w^i)$, where u^i is the i-th forward primitive query and $v^i\|w^i$ be the corresponding response or $v^i\|w^i$ be the i-th inverse primitive query and u^i be the corresponding response. Therefore, $\tau'_{\mathsf{P}} = \{(u^i, v^i\|w^i)\}_{i=1}^{q_s}$.

Additional Information Released to the Distinguisher. After the query-response phase, the distinguisher is provided additional information before outputting the decision bit. We want to mention that the additional information does not degrade the adversary's advantage as it is always possible to discard it.

For each construction query (x^i_c, z^i_c), $i = 1, \ldots, q_c$, the distinguisher will be given the query-response pairs of the forward primitive queries with $x^i_c\|c_1$ and $x^i_c\|c_2$. Thus, the modified transcript τ'' is of the form:

$$\tau'' = \{\tau'_c, \tau'_{\mathsf{P}} \cup \{(x^i_c\|c_1, v^i_{c_1}\|w^i_{c_1}), (x^i_c\|c_2, v^i_{c_2}\|w^i_{c_2})\}_{i=1}^{q_c}, b''\}$$

Bounding Distinguishing Advantage Between W_{sim} and W_{int}. Let the modified transcript equipped with additional information be denoted as $\tau'' = \tau''_{\mathsf{P}} \cup b''$, where $b'' \in \{\top, \bot\}$ represents the current status of the game after releasing additional information. It is easy to see that $b' = \bot$ means $b'' = \bot$ but even if $b' = \top$ it may happen that $b'' = \bot$. In the following, we define the following event:

$\mathsf{E} : b'' = \bot$ in τ''

Note that, the oracle in the intermediate world never aborts on any query, which implies that the probability of the event E holds in the intermediate world is zero. Therefore, if E occurs, the distinguisher identifies the simulator with a probability of 1. Therefore, we write

$$\begin{aligned} |\Pr[\mathcal{D}^{\mathsf{W}_{\text{sim}}} \to 1] - \Pr[\mathcal{D}^{\mathsf{W}_{\text{int}}} \to 1]| \quad &\leq \underbrace{|\Pr[\mathcal{D}^{\mathsf{W}_{\text{sim}}} \to 1|\overline{\mathsf{E}}] - \Pr[\mathcal{D}^{\mathsf{W}_{\text{int}}} \to 1|\overline{\mathsf{E}}]|}_{(\mathsf{A})} \\ &\quad + \Pr[\mathsf{E}] \end{aligned} \tag{16}$$

I. Upper Bounding $\Pr[\mathsf{E}]$**.** Note that the event E holds if the last l bits of the forward primitive query is either c_1 or c_2 and the response of the random function RF on the first $n-l$ bits of the queried string is of the form $z\|z\|0^{n-m}$, where $z \in \{0,1\}^m$. Since the output of the random function is uniformly distributed, one has

$$\Pr[\mathsf{E}] \leq \frac{2q}{2^n}, \tag{17}$$

where q_s is the number of simulator queries.

II. Upper Bounding (A). Since, the event E has not occurred so, $b'' = \top$ and we can omit b'' from the transcript. As the additional information enables the distinguisher to compute the responses corresponding to each construction query consistently, we can remove the construction queries from the transcript as well.[1] Now, the attack transcript is of the form

$$\tau = \tau'_{\mathsf{P}} \cup \{(x_c^i\|c_1, v_{c_1}^i\|w_{c_1}^i), (x_c^i\|c_2, v_{c_2}^i\|w_{c_2}^i)\}_{i=1}^{q_c}$$

Let $\mathbf{P}$ and $\mathbf{P}^*$ be the probabilities that the distinguisher $\mathcal{D}$ receives the transcript τ while interacting with the world W_{sim} and W_{int} respectively conditioned on E has not occurred. Suppose, $\mathbf{P}_i$ and $\mathbf{P}_i^*$ be the probabilities that i^{th} element of τ is received in the world W_{sim} and W_{int} respectively, given that the previous $(i-1)$ many pair have already been received. Let's first consider the world W_{sim}.

□ Case 1: i^{th} pair is generated for forward primitive query: In this case, $\mathcal{S}$ has invoked the algorithm $\mathcal{S}_{\mathsf{P}}^{+}$. Now since, $|\{v'\|w' \colon (*, v'\|w') \in \mathscr{D}\}| = (i-1)$, so, if $\text{right}_c(u) \neq c_1$ and $\text{right}_c(u) \neq c_2$, then we have

$$\mathbf{P}_i = \frac{1}{2^n - (i-1)}. \tag{18}$$

Otherwise, if $\text{right}_c(u) = c_1$, then we get

$$\mathbf{P}_i = \frac{1}{2^m} \cdot \frac{1}{2^{n-m} - (i-1)} = \frac{1}{2^n - 2^m \cdot (i-1)}, \tag{19}$$

using the randomness of z_1 and w_1. Similarly, if $\text{right}_c(u) = c_2$, then also we get the same bound for $\mathbf{P}_i$ using the randomness z_2 and z_3. Hence, from (18) and (19), overall we have

$$\mathbf{P}_i \geq \frac{1}{2^n - (i-1)}. \tag{20}$$

□ Case 2: i^{th} pair is generated for inverse primitive query: In this case, the algorithm $\mathcal{S}_{\mathsf{P}}^{-}$ has been invoked. Now, since $|(\{u' \colon (u', *) \in \mathscr{D}\} \cup \mathcal{C})| \geq (i-1)$. Hence, we have

$$\mathbf{P}_i \geq \frac{1}{2^n - (i-1)}. \tag{21}$$

Hence, from (20) and (21) we get that, whether the i-th query is forward or inverse in the world W_{id}, we always have

$$\mathbf{P}_i \geq \frac{1}{2^n - (i-1)}. \tag{22}$$

Now, we consider the world W_{int}, similar as the world W_{sim} here also we have following two cases:

[1] We can discard the construction query when considering sequential indifferentiability. However, this is not possible for regular indifferentiability, as discussed in [GBJ+23].

□ Case 1: i^{th} pair is generated for forward primitive query: In this case, the algorithm $\mathsf{P}_\star^{(+)}$ has been invoked. Since, the algorithm $\mathsf{P}_\star^{(+)}$ behaves exactly like a truly random permutation so, we have

$$\mathbf{P}_i^* = \frac{1}{2^n - (i-1)}. \tag{23}$$

□ Case 2: i^{th} pair is generated for inverse primitive query: In this case, the algorithm $\mathsf{P}_\star^{(-)}$ has been invoked. Now, since $|\mathcal{C}| = 2^{n-c+1}$ so,

$$(\{u\colon (u,*) \in \mathscr{D}_\star\} \cup \mathcal{C}) \leq (i-1) + 2^{n-l+1}.$$

Therefore, we have

$$\mathbf{P}_i^* \leq \frac{1}{2^n - (i-1) - 2^{n-l+1}}. \tag{24}$$

Hence, from (23) and (24) we get that, whether the i-th query is forward or inverse in the world W_{int}, we always have

$$\mathbf{P}_i^* \leq \frac{1}{2^n - (i-1) - 2^{n-l+1}}. \tag{25}$$

Now, using Eqn.(22) and Eqn.(25) we obtain

$$\begin{aligned}
\frac{\mathbf{P}}{\mathbf{P}^*} = \prod_{i=1}^{q_s} \frac{\mathbf{P}_i}{\mathbf{P}_i^*} &\geq \prod_{i=1}^{q_s} \frac{2^n - (i-1) - 2^{n-l+1}}{2^n - (i-1)} = \prod_{i=1}^{q_s} \left(1 - \frac{2^{n-l+1}}{2^n - (i-1)}\right) \\
&\geq \prod_{i=1}^{q_s} \left(1 - \frac{2^{n-l+1}}{2^n - q}\right) \geq \prod_{i=1}^{q_s} \left(1 - \frac{2^{n-l+1}}{2^n - 2^{n-1}}\right) \\
&\geq 1 - \sum_{i=1}^{q_s} \frac{2^{n-l+1}}{2^n - 2^{n-1}} = 1 - \sum_{i=1}^{q_s} \frac{1}{2^{l-2}} = 1 - \frac{q_s}{2^{l-2}} \geq 1 - \frac{q}{2^{l-2}}.
\end{aligned}$$

Therefore, from Theorem. (1) and Eqn.(26), we have,

$$\left|\Pr[\mathcal{D}^{\mathsf{W}_{\text{sim}}} \to 1|\overline{\mathsf{E}}] - \Pr[\mathcal{D}^{\mathsf{W}_{\text{int}}} \to 1|\overline{\mathsf{E}}]\right| \leq \frac{q}{2^{l-2}} \tag{26}$$

Finally, by combining Eqn. (16), Eqn. (17) and Eqn. (26), the result follows. □

5.3 Proof of Lemma 3

In this section, we demonstrate the success probability of the distinguisher, described in Fig. 8, against any simulator for the gSTH construction. Again, we recall the indifferentiability security definition:

$$\mathbf{Adv}^{\text{seq-indiff}}_{\text{gSTH},\mathcal{S}}(\mathcal{D}) = \left|\Pr[\mathcal{D}^{\mathsf{W}_{\text{sim}}} \to 0] - \Pr[\mathcal{D}^{\mathsf{W}_{\text{re}}} \to 0]\right|. \tag{27}$$

Bounding $\Pr[\mathcal{D}^{\mathsf{W}_{\text{sim}}} \to 0]$. By definition,

$$\Pr[\mathcal{D}^{\mathsf{W}_{\text{sim}}} \to 0] = 1 - \Pr[\mathcal{D}^{\mathsf{W}_{\text{sim}}} \to 1] \tag{28}$$

Note that $\mathcal{D}$ returns 1 only if, corresponding to some a_i, the simulator finds either $u_i = x_i \| c_1$ such that $\mathsf{RF}(x_i) = \text{left}_m(a_i) \| * \| *$, or $u_i = x_i \| c_2$ with $\mathsf{RF}(x_i) = * \| \text{left}_m(a_i) \| *$. For any $x \in \{0,1\}^{n-l}$, $\Pr[\mathsf{RF}(x) = \text{left}_m(a_i) \| * \| *] = \Pr[\mathsf{RF}(x) = * \| \text{left}_m(a_i) \| *] = 2^{-m}$. Since the runtime of any simulator is a polynomial of n, it can try at most $p(n)$ times to find such

u_i, where p is a polynomial. So, the probability of finding suitable $x_i\|c_1$ or $x_i\|c_2$ is upper bounded by $\frac{2p(n)}{2^m}$. Thus, from (28) we obtain

$$\Pr[\mathcal{D}^{\mathsf{W}_{\text{sim}}} \to 0] \geq 1 - \frac{2p(n)}{2^m}. \tag{29}$$

Bounding $\Pr[\mathcal{D}^{\mathsf{W}_{\text{re}}} \to 0]$. In this case, if $\mathcal{D}$ finds at least one $a_i \in \{0,1\}^n$ such that $\mathsf{P}^-(a_i) = *\|c_1$ or $\mathsf{P}^-(a_i) = *\|c_2$, then it outputs 1. So, the only way it outputs 0 is that for all a_i, $\mathsf{P}^-(a_i) \in \overline{\mathcal{C}}$, where $\mathcal{C} = \{x_i\|c_1 \colon x_i \in \{0,1\}^{n-l}\} \cup \{x_i\|c_2 \colon x_i \in \{0,1\}^{n-l}\}$. We have $|\overline{\mathcal{C}}| = 2^n - 2^{n-l+1}$. Now, from the randomness of permutation P and the freshness of each $v_i\|w_i$, we have $\Pr[\mathsf{P}^-(v_i\|w_i) \in \overline{\mathcal{C}}] \leq \frac{2^n - 2^{n-l+1} - (i-1)}{2^n - (i-1)}$, $\forall\ i \in [1, 2^l]$. Hence,

$$\begin{aligned}
\Pr[\mathcal{D}^{\mathsf{W}_{\text{re}}} \to 0] &= \Pr[\forall\ i \in [2^l],\ \mathsf{P}^-(v_i\|w_i) \in \overline{\mathcal{C}}] \leq \prod_{i=1}^{2^l} \frac{2^n - 2^{n-l+1} - (i-1)}{2^n - (i-1)} \\
&= \prod_{i=1}^{2^l} \left(1 - \frac{2^{n-l+1}}{2^n - (i-1)}\right) \leq \prod_{i=1}^{2^l} \left(1 - \frac{2^{n-l+1}}{2^n}\right) \\
&= \left(1 - \frac{1}{2^{l-1}}\right)^{2^l} \leq \frac{1}{e}.
\end{aligned} \tag{30}$$

By combining Eqn. (27), Eqn. (29), and Eqn. (30), we have

$$\mathbf{Adv}^{\text{seq-indiff}}_{\text{gSTH},\mathcal{S}}(\mathcal{D}) \geq \left|1 - \frac{2p(n)}{2^m} - \frac{1}{e}\right|. \tag{31}$$

5.4 Proof of Lemma 4

We first recall the events E_1 and E_2 as follows:

1. $\mathsf{E}_1 : b'' = \perp$ in τ''
2. $\mathsf{E}_2 : \exists \left(x^j_{p_1}, u^j_{p_1}, *\right) \in \tau''_{\mathsf{P}_1}, \left(v^i_{\mathsf{P}_2}, z^i_{\mathsf{P}_2}, *\right) \in \tau''_{\mathsf{P}_2}, \left(x^k_c, z^k_c\right) \in \tau'_c :$
   $x^j_{\mathsf{P}_1} \oplus u^j_{\mathsf{P}_1} = v^i_{\mathsf{P}_2} \wedge x^j_{\mathsf{P}_1} = x^k_c \wedge z^i_{\mathsf{P}_2} \neq z^k_c$

Now, we first upper bound the probability of the event E_1, and then we upper bound the probability of the event E_2 assuming that E_1 did not occur. Finally, we combine them to calculate the probability of the event E.

- **Upper Bounding** $\Pr[\mathsf{E}_1]$**:** This event will happen if any of the following three cases occur:

  □ Case 1: $\mathcal{S}^{\mathsf{P}_1(+)}$ **returns** $\perp$**:** Without loss of generality, let this occur in the i-th primitive query. This occurs only if the distinguisher can find x_i such that $\mathsf{RF}(x_i) = z_j$, for some $j < i$. Therefore, based on the randomness of RF, we have

  $$\Pr[\mathsf{E}_1] = \sum_{i=2}^{q_1} \sum_{j=1}^{i-1} \Pr[\mathsf{RF}(x_i) = z_j] \leq \sum_{i=2}^{q_1} \sum_{j=1}^{i-1} \frac{1}{2^n} = \sum_{i=2}^{q_1} \frac{i-1}{2^n} \leq \frac{q_1^2}{2^n}. \tag{32}$$

  □ Case 2: $\mathcal{S}^{\mathsf{P}_1(-)}$ **returns** $\perp$**:** This occurs if the simulator selects x_i such that $\mathsf{RF}(x_i)$ collides with some previous z_j. Therefore, employing a similar argument based on the randomness of the function RF, we have

  $$\Pr[\mathsf{E}_1] \leq \frac{q_1^2}{2^n}. \tag{33}$$

□ Case 3: $\mathcal{S}^{\mathsf{P}_2(+)}$ **returns** $\perp$**:** Using similar argument as Case 2, we will have

$$\Pr[\mathsf{E}_1] \leq \frac{q_2^2}{2^n}. \tag{34}$$

By combining Eqn. (32), Eqn. (33), and Eqn. (34), and $q = q_1 + q_2$, we have

$$\Pr[\mathsf{E}_1] \leq \frac{3q^2}{2^n}. \tag{35}$$

- **Upper Bounding** $\Pr[\mathsf{E}_2|\overline{\mathsf{E}_1}]$**:** Note that, $\overline{\mathsf{E}_1}$ implies that no two different construction queries resulted in the same z value. Moreover, our design of the simulator ensures that

  1. $(x_{\mathsf{P}_1}, u_{\mathsf{P}_1}), (x'_{\mathsf{P}_1}, u'_{\mathsf{P}_1}) \in \tau''_{\mathsf{P}_1} \implies (x_{\mathsf{P}_1} \neq x'_{\mathsf{P}_1}) \wedge (u_{\mathsf{P}_1} \neq u'_{\mathsf{P}_1})$.
  2. $(v_{\mathsf{P}_2}, z_{\mathsf{P}_2}), (v'_{\mathsf{P}_2}, z'_{\mathsf{P}_2}) \in \tau''_{\mathsf{P}_2} \implies (v_{\mathsf{P}_2} \neq v'_{\mathsf{P}_2}) \wedge (z_{\mathsf{P}_2} \neq z'_{\mathsf{P}_2})$.

  Furthermore, due to the design of the simulator algorithms, $\forall \ \left(x^j_{p_1}, u^j_{p_1}, *\right) \in \tau''_{\mathsf{P}_1}$, there exists exactly one $\left(v^i_{\mathsf{P}_2}, z^i_{\mathsf{P}_2}, *\right) \in \tau''_{\mathsf{P}_2}$ such that $(x^j_{\mathsf{P}_1} \oplus u^j_{\mathsf{P}_1} = v^i_{\mathsf{P}_2}) \wedge (z^i_{\mathsf{P}_2} = \mathsf{RF}(x^j_{p_1}))$. Therefore, if $x^k_c = x^j_{\mathsf{P}_1}$, then $z^i_{\mathsf{P}_2} = \mathsf{RF}(x^j_{p_1}) = \mathsf{RF}(x^k_c) = z^k_c$. Hence, we have

$$\Pr[\mathsf{E}_2] = 0. \tag{36}$$

Finally, by combining Eqn. (35) and Eqn. (36), we have

$$\Pr[\mathsf{E}] \leq \frac{3q^2}{2^n}. \tag{37}$$

5.5 Proof of Lemma 5

Since the event $\mathsf{E} = \mathsf{E}_1 \vee \mathsf{E}_2$ has not occurred, all the primitive queries corresponding to τ'_c are present in the transcript $\tau''_{\mathsf{P}_1} \cup \tau''_{\mathsf{P}_2}$, and due to $\overline{\mathsf{E}_2}$, there is no inconsistency between primitive and construction queries. Consequently, we consider τ'_c as redundant, and we can discard the query-response pair in τ'_c.[2] Additionally, due to $\overline{\mathsf{E}_1}$, we know that $b'' = \top$, so we will also discard b'' from the transcript. Thus, updated transcript is as follows: $\tau = \{\tau''_{\mathsf{P}_1}, \tau''_{\mathsf{P}_2}\}$. We apply the H-Coefficient technique with respect to the transcript τ to upper bound the distinguishing advantage of $\mathcal{D}$ between the real and the simulated world, conditioned on E has not occurred. Note that, due to $\overline{\mathsf{E}}$, there is no permutation incompatibility or inconsistency between primitive and construction queries in any transcript τ obtained by interacting with the simulator. Since there are no bad events with respect to the transcript τ, the only task is to lower bound the ratio of the simulated to real world interpolation probability for a good transcript. Let,

$$\tau = \{(x_1, u_1, s_1), \ldots, (x_{q_1}, u_{q_1}, s_{q_1}), (v_1, z_1, s_1), \ldots, (v_{q_2}, z_{q_2}, s_{q_2})\}$$

be the transcript, where i-th triple in τ implies $\mathcal{D}$ received the triple during the i-th forward or inverse query to the primitive oracles in the simulator world or in the real world.

Let $\mathbf{P}$ and $\mathbf{P}^*$ be the probabilities that $\mathcal{D}$ receives the transcript τ while interacting with the oracles in the real world or the simulated world, respectively. Without loss of

[2] We can discard the construction query when considering sequential indifferentiability. However, this is not possible for regular indifferentiability, as discussed in [GBJ+23].

generality, we assume that $\mathcal{D}$ does not make any duplicate queries. Therefore, in the real world, we have,

$$\mathbf{P} \leq \prod_{i=1}^{q} \frac{1}{2^n - (i-1)},\ q = q_1 + q_2. \tag{38}$$

To lower bound $\mathbf{P}^*$, let $\mathbf{P}_i^*$ be the probability that the i-th triple of τ is received, given that the previous $(i-1)$ triples have already been received.

□ Case 1: i-th triple is a forward query to $\mathcal{S}^{\mathsf{P}_1(+)}$: In this case, the algorithm $\mathcal{S}^{\mathsf{P}_1(+)}$ was invoked. Here, u is chosen randomly from the set

$$\{0,1\}^n \setminus \left(\{u' : (*,u',*) \in \mathscr{D}_1\} \bigcup \{x \oplus v' : (v',*,*) \in \mathscr{D}_2\}\right).$$

Moreover, we have $|\{u' : (*,u',*) \in \mathscr{D}_1\} \bigcup \{x \oplus v' : (v',*,*) \in \mathscr{D}_2\}| \geq (i-1)$. Therefore,

$$\mathbf{P}_i^* \geq \frac{1}{2^n - (i-1)}. \tag{39}$$

□ Case 2: i-th triple is an inverse query to $\mathcal{S}^{\mathsf{P}_1(-)}$: In this case, the algorithm $\mathcal{S}^{\mathsf{P}_1(-)}$ was invoked. Here, x is chosen randomly from the set

$$\{0,1\}^n \setminus \left(\{x' : (x',*,*) \in \mathscr{D}_1\} \bigcup \{u \oplus v' : (v',*,*) \in \mathscr{D}_2\}\right).$$

Therefore,

$$\mathbf{P}_i^* \geq \frac{1}{2^n - (i-1)}. \tag{40}$$

□ Case 3: i-th triple is a forward query to $\mathcal{S}^{\mathsf{P}_2(+)}$: For this case, $\mathcal{S}$ invoked the algorithm $\mathcal{S}^{\mathsf{P}_2(+)}$ and the output value z is the output of the random function RF. Therefore,

$$\mathbf{P}_i^* \geq \frac{1}{2^n}, \tag{41}$$

where the term 2^{-n} comes from the probability of the random function.

□ Case 4: i^{th} triple is an inverse query to $\mathcal{S}^{\mathsf{P}_2(-)}$: For this case, $\mathcal{S}$ invoked the algorithm $\mathcal{S}^{\mathsf{P}_2(-)}$. Here, v is chosen randomly from $\{0,1\}^n \setminus (\{v' : (v',*,*) \in \mathscr{D}_2\})$. Therefore,

$$\mathbf{P}_i^* \geq \frac{1}{2^n - (i-1)}. \tag{42}$$

Hence, by following Eqn. (39)-(42), whether the i-th query is forward or inverse to $\mathcal{S}^{\mathsf{P}_1}$ or $\mathcal{S}^{\mathsf{P}_2}$, we always have,

$$\mathbf{P}_i^* \geq \frac{1}{2^n} = \frac{\left(1 - \frac{i-1}{2^n}\right)}{2^n - (i-1)}. \tag{43}$$

Using Eqn. (38) and Eqn. (43), we have,

$$\mathbf{P}^* = \prod_{i=1}^{q} \mathbf{P}_i^* \geq \prod_{i=1}^{q} \frac{\left(1 - \frac{i-1}{2^n}\right)}{2^n - (i-1)} \geq \mathbf{P} \cdot \prod_{i=1}^{q} \left(1 - \frac{i-1}{2^n}\right) \geq \mathbf{P} \cdot \left(1 - \frac{q^2}{2^n}\right). \tag{44}$$

Therefore, by following Thm. 1 and Eqn. (44), we have,

$$|\Pr[\mathcal{D}^{\mathsf{W}_{\mathrm{sim}}} \to 1|\overline{\mathsf{E}}] - \Pr[\mathcal{D}^{\mathsf{W}_{\mathrm{re}}} \to 1|\overline{\mathsf{E}}]| \leq \frac{q^2}{2^n}. \tag{45}$$

6 Conclusion

In this paper, we have studied the indifferentiability security of two block cipher based PRF constructions, namely STH and EDM constructions in the sequential model. We have shown that STH is not indifferentiable from a public random function, as we have demonstrated a distinguisher that distinguishes the construction from an ideal function in constant number of queries. Next, we have shown that a generalized version of the STH construction, dubbed as gSTH, is indifferentiable from a public random function up to 2^l many queries. We have also shown that our proven bound is tight irrespective of any simulator. Next, we have shown that EDM achieves $n/2$-bit indifferentiable bound, and our proven bound is tight in terms of our proposed simulator. Moreover, we have also shown that the single permutation variant of EDM is not indifferentiable from a public random function.

References

[BCG+12] Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Søren S. Thomsen, and Tolga Yalçin. PRINCE - A low-latency block cipher for pervasive computing applications - extended abstract. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, volume 7658 of *Lecture Notes in Computer Science*, pages 208–225. Springer, 2012. doi:10.1007/978-3-642-34961-4_14.

[BI99] Mihir Bellare and Russell Impagliazzo. A tool for obtaining tighter security analyses of pseudorandom function based constructions, with applications to PRP to PRF conversion, 1999. URL: http://eprint.iacr.org/1999/024.

[BKL+07] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: an ultra-lightweight block cipher. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer, 2007. doi:10.1007/978-3-540-74735-2_31.

[BKR98] Mihir Bellare, Ted Krovetz, and Phillip Rogaway. Luby-rackoff backwards: Increasing security by making block ciphers non-invertible. In Kaisa Nyberg, editor, *Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31 - June 4, 1998, Proceeding*, volume 1403 of *Lecture Notes in Computer Science*, pages 266–280. Springer, 1998. URL: https://doi.org/10.1007/BFb0054132, doi:10.1007/BFB0054132.

[BN18] Srimanta Bhattacharya and Mridul Nandi. Full indifferentiable security of the xor of two or more random permutations using the \chi ^2 method. In Jesper Buus Nielsen and Vincent Rijmen, editors, *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*, volume 10820 of *Lecture Notes in Computer Science*, pages 387–412. Springer, 2018. doi:10.1007/978-3-319-78381-9_15.

[BPP+17] Subhadeep Banik, Sumit Kumar Pandey, Thomas Peyrin, Yu Sasaki, Siang Meng Sim, and Yosuke Todo. GIFT: A small present - towards reaching the limit of lightweight encryption. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, volume 10529 of *Lecture Notes in Computer Science*, pages 321–345. Springer, 2017. doi:10.1007/978-3-319-66787-4_16.

[CDMP05] Jean-Sébastien Coron, Yevgeniy Dodis, Cécile Malinaud, and Prashant Puniya. Merkle-damgård revisited: How to construct a hash function. In Victor Shoup, editor, *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, volume 3621 of *Lecture Notes in Computer Science*, pages 430–448. Springer, 2005. doi:10.1007/11535218_26.

[CDNY18] Avik Chakraborti, Nilanjan Datta, Mridul Nandi, and Kan Yasuda. Beetle family of lightweight and secure authenticated encryption ciphers. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(2):218–241, 2018. URL: https://doi.org/10.13154/tches.v2018.i2.218-241, doi:10.13154/TCHES.V2018.I2.218-241.

[CEJS23] Benoît Cogliati, Jordan Ethan, Ashwin Jha, and Soumya Kanti Saha. On large tweaks in tweakable even-mansour with linear tweak and key mixing. *IACR Trans. Symmetric Cryptol.*, 2023(4):330–364, 2023. URL: https://doi.org/10.46586/tosc.v2023.i4.330-364, doi:10.46586/TOSC.V2023.I4.330-364.

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In Jeffrey Scott Vitter, editor, *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 209–218. ACM, 1998. doi:10.1145/276698.276741.

[CLL19] Wonseok Choi, ByeongHak Lee, and Jooyoung Lee. Indifferentiability of truncated random permutations. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part I*, volume 11921 of *Lecture Notes in Computer Science*, pages 175–195. Springer, 2019. doi:10.1007/978-3-030-34578-5_7.

[CN08] Donghoon Chang and Mridul Nandi. A short proof of the PRP/PRF switching lemma. *IACR Cryptol. ePrint Arch.*, page 78, 2008. URL: http://eprint.iacr.org/2008/078.

[CPS08] Jean-Sébastien Coron, Jacques Patarin, and Yannick Seurin. The random oracle model and the ideal cipher model are equivalent. In David A. Wagner, editor, *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings*, volume 5157 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2008. doi:10.1007/978-3-540-85174-5_1.

[CS15] Benoit Cogliati and Yannick Seurin. On the provable security of the iterated even-mansour cipher against related-key and chosen-key attacks. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications*

of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I, volume 9056 of *Lecture Notes in Computer Science*, pages 584–613. Springer, 2015. doi:10.1007/978-3-662-46800-5_23.

[CS16a] Benoît Cogliati and Yannick Seurin. EWCDM: an efficient, beyond-birthday secure, nonce-misuse resistant MAC. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 121–149. Springer, 2016. doi:10.1007/978-3-662-53018-4_5.

[CS16b] Benoît Cogliati and Yannick Seurin. Strengthening the known-key security notion for block ciphers. In Thomas Peyrin, editor, *Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers*, volume 9783 of *Lecture Notes in Computer Science*, pages 494–513. Springer, 2016. doi:10.1007/978-3-662-52993-5_25.

[DDNP18] Nilanjan Datta, Avijit Dutta, Mridul Nandi, and Goutam Paul. Double-block hash-then-sum: A paradigm for constructing BBB secure PRF. *IACR Trans. Symmetric Cryptol.*, 2018(3):36–92, 2018. URL: https://doi.org/10.13154/tosc.v2018.i3.36-92, doi:10.13154/TOSC.V2018.I3.36-92.

[DDNY18] Nilanjan Datta, Avijit Dutta, Mridul Nandi, and Kan Yasuda. Encrypt or decrypt? to make a single-key beyond birthday secure nonce-based MAC. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part I*, volume 10991 of *Lecture Notes in Computer Science*, pages 631–661. Springer, 2018. doi:10.1007/978-3-319-96884-1_21.

[DHT17] Wei Dai, Viet Tung Hoang, and Stefano Tessaro. Information-theoretic indistinguishability via the chi-squared method. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III*, volume 10403 of *Lecture Notes in Computer Science*, pages 497–523. Springer, 2017. doi:10.1007/978-3-319-63697-9_17.

[DNS22] Avijit Dutta, Mridul Nandi, and Abishanka Saha. Proof of mirror theory for $\xi_{\max} = 2$. *IEEE Trans. Inf. Theory*, 68(9):6218–6232, 2022. doi:10.1109/TIT.2022.3171178.

[DPP08] Yevgeniy Dodis, Krzysztof Pietrzak, and Prashant Puniya. A new mode of operation for block ciphers and length-preserving macs. In Nigel P. Smart, editor, *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*, volume 4965 of *Lecture Notes in Computer Science*, pages 198–219. Springer, 2008. doi:10.1007/978-3-540-78967-3_12.

[DR98] Joan Daemen and Vincent Rijmen. The block cipher rijndael. In Jean-Jacques Quisquater and Bruce Schneier, editors, *Smart Card Research and Applications, This International Conference, CARDIS '98, Louvain-la-Neuve, Belgium, September 14-16, 1998, Proceedings*, volume 1820 of *Lecture Notes in Computer Science*, pages 277–284. Springer, 1998. doi:10.1007/10721064_26.

[DRRS09] Yevgeniy Dodis, Leonid Reyzin, Ronald L. Rivest, and Emily Shen. Indifferentiability of permutation-based compression functions and tree-based modes of operation, with applications to MD6. In Orr Dunkelman, editor, *Fast Software Encryption, 16th International Workshop, FSE 2009, Leuven, Belgium, February 22-25, 2009, Revised Selected Papers*, volume 5665 of *Lecture Notes in Computer Science*, pages 104–121. Springer, 2009. doi:10.1007/978-3-642-03317-9_7.

[GBJ+23] Aldo Gunsing, Ritam Bhaumik, Ashwin Jha, Bart Mennink, and Yaobin Shen. Revisiting the indifferentiability of the sum of permutations. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part III*, volume 14083 of *Lecture Notes in Computer Science*, pages 628–660. Springer, 2023. doi:10.1007/978-3-031-38548-3_21.

[GG21] Shoni Gilboa and Shay Gueron. The advantage of truncated permutations. *Discret. Appl. Math.*, 294:214–223, 2021. URL: https://doi.org/10.1016/j.dam.2021.01.029, doi:10.1016/J.DAM.2021.01.029.

[GGM84] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions (extended abstract). In *25th Annual Symposium on Foundations of Computer Science, West Palm Beach, Florida, USA, 24-26 October 1984*, pages 464–479. IEEE Computer Society, 1984. doi:10.1109/SFCS.1984.715949.

[GM20] Aldo Gunsing and Bart Mennink. The summation-truncation hybrid: Reusing discarded bits for free. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part I*, volume 12170 of *Lecture Notes in Computer Science*, pages 187–217. Springer, 2020. doi:10.1007/978-3-030-56784-2_7.

[GSWG19] Chun Guo, Yaobin Shen, Lei Wang, and Dawu Gu. Beyond-birthday secure domain-preserving prfs from a single permutation. *Des. Codes Cryptogr.*, 87(6):1297–1322, 2019. URL: https://doi.org/10.1007/s10623-018-0528-8, doi:10.1007/S10623-018-0528-8.

[Gun22] Aldo Gunsing. Block-cipher-based tree hashing. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part IV*, volume 13510 of *Lecture Notes in Computer Science*, pages 205–233. Springer, 2022. doi:10.1007/978-3-031-15985-5_8.

[HWKS98] Chris Hall, David A. Wagner, John Kelsey, and Bruce Schneier. Building prfs from prps. In Hugo Krawczyk, editor, *Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings*, volume 1462 of *Lecture Notes in Computer Science*, pages 370–389. Springer, 1998. URL: https://doi.org/10.1007/BFb0055742, doi:10.1007/BFB0055742.

[Iwa06] Tetsu Iwata. New blockcipher modes of operation with beyond the birthday bound security. In Matthew J. B. Robshaw, editor, *Fast Software Encryption, 13th International Workshop, FSE 2006, Graz, Austria, March 15-17, 2006, Revised Selected Papers*, volume 4047 of *Lecture Notes in Computer Science*, pages 310–327. Springer, 2006. doi:10.1007/11799313_20.

[KR07] Lars R. Knudsen and Vincent Rijmen. Known-key distinguishers for some block ciphers. In Kaoru Kurosawa, editor, *Advances in Cryptology - ASIACRYPT 2007, 13th International Conference on the Theory and Application of Cryptology and Information Security, Kuching, Malaysia, December 2-6, 2007, Proceedings*, volume 4833 of *Lecture Notes in Computer Science*, pages 315–324. Springer, 2007. doi:10.1007/978-3-540-76900-2_19.

[Lee17] Jooyoung Lee. Indifferentiability of the sum of random permutations toward optimal security. *IEEE Trans. Inf. Theory*, 63(6):4050–4054, 2017. doi:10.1109/TIT.2017.2679757.

[LS13] Rodolphe Lampe and Yannick Seurin. How to construct an ideal cipher from a small set of public permutations. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part I*, volume 8269 of *Lecture Notes in Computer Science*, pages 444–463. Springer, 2013. doi:10.1007/978-3-642-42033-7_23.

[MN17] Bart Mennink and Samuel Neves. Optimal prfs from blockcipher designs. *IACR Trans. Symmetric Cryptol.*, 2017(3):228–252, 2017. URL: https://doi.org/10.13154/tosc.v2017.i3.228-252, doi:10.13154/TOSC.V2017.I3.228-252.

[MP15] Bart Mennink and Bart Preneel. On the XOR of multiple random permutations. In Tal Malkin, Vladimir Kolesnikov, Allison Bishop Lewko, and Michalis Polychronakis, editors, *Applied Cryptography and Network Security - 13th International Conference, ACNS 2015, New York, NY, USA, June 2-5, 2015, Revised Selected Papers*, volume 9092 of *Lecture Notes in Computer Science*, pages 619–634. Springer, 2015. doi:10.1007/978-3-319-28166-7_30.

[MPN10] Avradip Mandal, Jacques Patarin, and Valérie Nachef. Indifferentiability beyond the birthday bound for the xor of two public random permutations. In Guang Gong and Kishan Chand Gupta, editors, *Progress in Cryptology - INDOCRYPT 2010 - 11th International Conference on Cryptology in India, Hyderabad, India, December 12-15, 2010. Proceedings*, volume 6498 of *Lecture Notes in Computer Science*, pages 69–81. Springer, 2010. doi:10.1007/978-3-642-17401-8_6.

[MPS12] Avradip Mandal, Jacques Patarin, and Yannick Seurin. On the public indifferentiability and correlation intractability of the 6-round feistel construction. In Ronald Cramer, editor, *Theory of Cryptography - 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy, March 19-21, 2012. Proceedings*, volume 7194 of *Lecture Notes in Computer Science*, pages 285–302. Springer, 2012. doi:10.1007/978-3-642-28914-9_16.

[MRH04] Ueli M. Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In Moni Naor, editor, *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004, Proceedings*, volume 2951 of *Lecture Notes in Computer Science*, pages 21–39. Springer, 2004. doi:10.1007/978-3-540-24638-1_2.

[RSS11] Thomas Ristenpart, Hovav Shacham, and Thomas Shrimpton. Careful with composition: Limitations of the indifferentiability framework. In Kenneth G. Paterson, editor, *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual*

International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings, volume 6632 of *Lecture Notes in Computer Science*, pages 487–506. Springer, 2011. doi:10.1007/978-3-642-20465-4_27.

[WC81] Mark N. Wegman and Larry Carter. New hash functions and their use in authentication and set equality. *J. Comput. Syst. Sci.*, 22(3):265–279, 1981. doi:10.1016/0022-0000(81)90033-7.