Indifferentiability of Truncated Random Permutations

Wonseok Choi, Byeonghak Lee, and Jooyoung Lee*

{krwioh,lbh0307,hicalf}@kaist.ac.kr

KAIST, Korea

Abstract. One of natural ways of constructing a pseudorandom function from a pseudorandom permutation is to simply truncate the output of the permutation. When n is the permutation size and m is the number of truncated bits, the resulting construction is known to be indistinguishable from a random function up to $2^{\frac{n+m}{2}}$ queries, which is tight.
In this paper, we study the indifferentiability of a truncated random permutation where a fixed prefix is prepended to the inputs. We prove that this construction is (regularly) indifferentiable from a public random function up to $\min\{2^{\frac{n+m}{3}}, 2^m, 2^\ell\}$ queries, while it is publicly indifferentiable up to $\min\{\max\{2^{\frac{n+m}{3}}, 2^{\frac{n}{2}}\}, 2^\ell\}$ queries, where ℓ is the size of the fixed prefix. Furthermore, the regular indifferentiability bound is proved to be tight when $m + \ell \ll n$.
Our results significantly improve upon the previous bound of $\min\{2^{\frac{m}{2}}, 2^\ell\}$ given by Dodis et. al (FSE 2009), allowing us to construct, for instance, an $\frac{n}{2}$-to-$\frac{n}{2}$ bit random function that makes a single call to an n-bit permutation, achieving $\frac{n}{2}$-bit security.

Keywords: random permutation, random function, truncation, indifferentiability, chi-square method

1 Introduction

A block cipher is typically modeled as a pseudorandom permutation in a provable security setting: no distinguisher should be able to distinguish the block cipher from a truly random permutation by making a certain number of encryption and decryption queries in a black-box manner. However, for some modes of operation, one might want the block cipher to behave like a pseudorandom function. A variety of cryptographic protocols (such as signature schemes, random number generators, key derivation schemes, etc.) provide provable security in the random oracle model. This observation motivates the problem of constructing a pseudorandom function from pseudorandom permutations. Sometimes this

* Jooyoung Lee was supported by a National Research Foundation of Korea (NRF) grant funded by the Korean government (Ministry of Science and ICT), No. NRF-2017R1E1A1A03070248.

problem is called "Luby-Rackoff backward" [2]: the Feistel network transforms a set of (not necessarily one-to-one) functions into a permutation, and this problem considers its opposite direction. In this direction, two approaches are natural and straightforward; one is to xor multiple independent random permutations and the other is to simply truncate the output of the permutation.

In this work, we will focus on the security of a truncated random permutation. One advantage of this construction (over xoring multiple permutations) is its minimality; it is based on a single permutation, using only a single call to the permutation. We will study the security of a truncated random permutation in the indifferentiability framework. In this framework, we will fix some of the input bits to the permutation, since otherwise one can easily differentiate the construction from a public random function F by making a backward query v to the simulator S, and then checking out if $\mathsf{F}(\mathsf{S}^{-1}(v)) = v$. Later we will discuss this attack in more detail.

Truncated Permutation. Let n, ℓ, m be positive integers such that $\ell, m < n$. Our construction is precisely defined as

$$\mathsf{TRP}[\mathsf{P}] \stackrel{\text{def}}{=} \mathsf{Tr}_m(\mathsf{P}(c \,\|\, \cdot)),$$

where $c \in \{0,1\}^\ell$ is an ℓ-bit prefix, P is an n-bit permutation (modeled as a random permutation oracle), and

$$\begin{aligned} \mathsf{Tr}_m : \{0,1\}^n &\longrightarrow \{0,1\}^{n-m} \\ x &\longmapsto x_R, \end{aligned}$$

when $x \in \{0,1\}^n$ is written as $x_L \,\|\, x_R$ for $x_L \in \{0,1\}^m$ and $x_R \in \{0,1\}^{n-m}$. (So Tr_m truncates the first m bits of the input.) In this way, we obtain an $(n-\ell)$-to-$(n-m)$ bit function from an n-bit permutation.

In order to prove that this construction is indifferentiable from a public random function F, one should present a simulator S that emulates P having access to F so that it is infeasible to distinguish two systems $(\mathsf{F}, \mathsf{S}[\mathsf{F}])$ and $(\mathsf{TRP}[\mathsf{P}], \mathsf{P})$.

As far as we know, the indifferentiability of TRP has been studied only in [6], where the adversarial differentiating advantage is upper bounded by

$$\frac{(q_F + q_S)^2}{2^n} + \frac{q_F q_S}{2^m} + \frac{q_S}{2^\ell},$$

where q_F and q_S denote the number of function queries and the number of simulator queries, respectively.

Our Contribution. In the indifferentiability framework, we consider two different notions; (regular) indifferentiability and public indifferentiability. With respect to regular indifferentiability, we present a simulator S such that any distinguisher is able to distinguish $(\mathsf{F}, \mathsf{S}[\mathsf{F}])$ and $(\mathsf{TRP}[\mathsf{P}], \mathsf{P})$ with probability at most

$$\left(\frac{(q_F + q_S)^3}{2^{n+m-1}}\right)^{\frac{1}{2}} + \frac{(3\ln q_F + 3(n-m) + 1)q_S}{2^{m-1}} + \frac{5q_S}{2^{\ell-1}}.$$

We also prove that the regular indifferentiability bound is tight when $m+\ell \ll n$.

With respect to public indifferentiability, we present a simulator S such that any distinguisher is able to distinguish $(\mathsf{F}, \mathsf{S}[\mathsf{F}])$ and $(\mathsf{TRP}[\mathsf{P}], \mathsf{P})$ with probability at most

$$\left(\frac{(q_F+q_S)^3}{2^{n+m-1}}\right)^{\frac{1}{2}} + \frac{q_S}{2^{\ell-1}}$$

if $q_F+q_S < 2^m$, and

$$\left(\frac{5(q_F+q_S)^2}{2^{n+1}}\right)^{\frac{1}{2}} + \frac{q_S}{2^{\ell-1}},$$

otherwise. Figure 1 compares our bounds and the bound from [6] in terms of the threshold number of queries q (in log base 2), where $q = q_F + q_S$; TRP is regularly indifferentiable (resp. publicly indifferentiable) from a public random function up to $\min\{2^{\frac{n+m}{3}}, 2^m, 2^\ell\}$ (resp. $\min\{\max\{2^{\frac{n+m}{3}}, 2^{\frac{n}{2}}\}, 2^\ell\}$) queries, improving upon the previous bound of $\min\{2^{\frac{m}{2}}, 2^\ell\}$.

Our results allow us to construct an n-to-n bit random function that makes a single call to a wider $2n$-bit permutation, achieving n-bit security. This construction is comparable to the sum of two independent permutations, $\mathsf{P}_1 \oplus \mathsf{P}_2$, that makes two calls to the underlying n-bit permutations P_1 and P_2 to achieve n-bit security. For each simulator query, our simulator makes at most one call to the public random function F, while the simulator for $\mathsf{P}_1 \oplus \mathsf{P}_2$ (given in [3]) might possibly make n calls to F.

By letting $q_S = 0$, an indifferentiability bound of TRP is reduced to an indistinguishability bound of TRP. Without any simulator query, we can make our computation even tighter, recovering the optimal indistinguishability bound of TRP given in [8]. See Appendix A.

We remark that efficient and secure construction of a fixed-input-length random oracle (FIL-RO) can be of practical relevance. As a FIL-RO, TRP founds various applications; a public finalization function for MACs, a non-compressing primitive for compression functions [21], a key derivation function, etc. A key derivation function in GCM-SIV was also proposed to use TRP [9, 10], although later studies offered alternatives [12,21]. We already have large and secure permutations at hand, including KECCAK and GIMLI, that can be used to construct a FIL-RO with reasonable size and security.

Related Work. The sum of two random permutations was first considered by Bellare et al. [2] in the indistinguishability framework. Subsequently, a series of works improved this seminal result [1, 4, 14, 19, 20], culminating with the proof by Dai et al. [5] that the sum of two n-bit random permutations is (fully) secure up to 2^n queries.

In the indifferentiability model, Mandal et al. [15] proved that the sum of two public random permutations is secure up to $2^{\frac{2n}{3}}$ queries, and later Mennink and Preneel [19] pointed out a flaw in their security proof and fixed it. Lee [13] proved that the sum of k independent random permutations is secure up to $2^{\frac{(k-1)n}{k}}$ queries. Finally, Bhattacharya and Nandi [3] proved that the sum of two random permutations is secure up to 2^n queries.

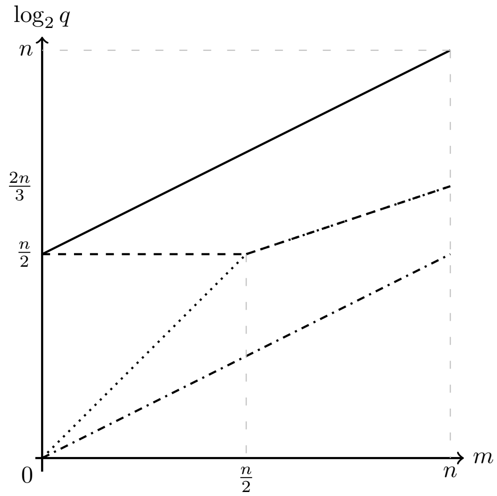

Fig. 1: Our regular and public indifferentiability bounds for TRP as a function of m (ignoring ℓ). For all parameters below the dashed line (resp. the dotted line), TRP is regularly indifferentiable (resp. publicly indifferentiable) from a public random function. The solid and dash-dotted lines represent the indistinguishability bound [8] and the previous indifferentiability bound [6], respectively.

Truncating a random permutation was first considered by Hall et al. [11], where they proved the security of TRP (with $\ell = 0$) up to $\min\{2^{\frac{n+m}{2}}, 2^{\frac{2(n-m)}{3}}\}$ queries in terms of indistinguishability. Bellare and Impagliazzo [1] improved this bound up to $\min\{2^{2m}, 2^{\frac{n+m}{2}}\}$. Recently, Gilboa et al. [8] proved that TRP is indistinguishable from a random function up to $2^{\frac{n+m}{2}}$ queries. This bound turns out to be tight as they also present matching attacks. Mennink [18] generalized truncation functions used in TRP, and showed that the security of such constructions (in terms of indistinguishability) cannot exceed that of the original TRP.

As mentioned before, Dodis et al. [6] proved the security of TRP up to $\min\{2^{\frac{m}{2}}, 2^{\ell}\}$ queries in terms of indifferentiability, and used it to build the MD6 hash function. Precisely, the MD6 hash function uses TRP with $n = 5696$, $\ell = 960$ and $m = 4672$.

2 Preliminaries

Notation. Throughout this work, we fix positive integers n, m, ℓ such that $m, \ell < n$ to denote the size of the underlying permutation P, the number of truncated bits and the prefix size of TRP, respectively. We also fix $c \in \{0,1\}^{\ell}$ to denote the prefix of TRP. We will write $\mathcal{C} = \{c \,\|\, x : x \in \{0,1\}^{n-\ell}\}$.

Regular and Public Indifferentiability. In the indifferentiability framework, a distinguisher is given two systems $(\mathsf{C}[\mathsf{P}], \mathsf{P})$ and $(\mathsf{F}, \mathsf{S}[\mathsf{F}])$, where P is an ideal primitive, $\mathsf{C}[\mathsf{P}]$ is a bigger construction using P as a building block, F is another ideal primitive with the same interface as $\mathsf{C}[\mathsf{P}]$, and $\mathsf{S}[\mathsf{F}]$ is a probabilistic Turing machine with the same interface as P that has oracle access to F. The goal of the *simulator* $\mathsf{S}[\mathsf{F}]$ is to emulate the ideal primitive P so that no distinguisher can tell apart the two systems $(\mathsf{F}, \mathsf{S}[\mathsf{F}])$ and $(\mathsf{C}[\mathsf{P}], \mathsf{P})$ with a significant probability, based on their responses to queries that the distinguisher may send. We say that the construction $\mathsf{C}[\mathsf{P}]$ is indifferentiable from the ideal primitive F if the existence of such a simulator is proved. The indifferentiability guarantees universal composability of $\mathsf{C}[\mathsf{P}]$: if $\mathsf{C}[\mathsf{P}]$ is indifferentiable from F, then $\mathsf{C}[\mathsf{P}]$ can replace F in any cryptosystem, and the resulting cryptosystem is at least as secure under the assumption that P is ideal as under the assumption that F is ideal.

More precisely, in an information-theoretic sense, a construction C with oracle access to an ideal primitive P is said to be (q_F, q_S, ε)*-regular indifferentiable from an ideal primitive* F if there exists a simulator S with oracle access to F such that for any distinguisher $\mathcal{A}$ making exactly q_F queries to the outer construction ($\mathsf{C}[\mathsf{P}]$ or F) and exactly q_S queries to the inner primitive (P or $\mathsf{S}[\mathsf{F}]$),[1] it holds that

$$\mathbf{Adv}^{\mathsf{reg}}_{\mathsf{C},\mathsf{S}}(\mathcal{A}) \stackrel{\text{def}}{=} \left|\Pr\left[1 \leftarrow \mathcal{A}^{\mathsf{C}[\mathsf{P}],\mathsf{P}}\right] - \Pr\left[1 \leftarrow \mathcal{A}^{\mathsf{F},\mathsf{S}[\mathsf{F}]}\right]\right| < \varepsilon.$$

See [17] for more detail on indifferentiability.

Public indifferentiability has been introduced in [7,22] and formalized in [16] as a variant of indifferentiability, where the simulator knows all queries made by the distinguisher to the primitive it tries to simulate. This weaker notion is useful to argue the security of cryptosystems where all the queries to the ideal primitive are public (as e.g., in many digital signature schemes). The adversarial public-differentiating advantage $\mathbf{Adv}^{\mathsf{pub}}_{\mathsf{C},\mathsf{S}}(\mathcal{A})$ is similarly defined for any distinguisher $\mathcal{A}$, and hence (q_F, q_S, ε)*-public indifferentiability.*

The χ^2 Method. We give here all the necessary background on the χ^2 method [5] that we will use throughout this paper.

We fix a set of random systems, a deterministic distinguisher $\mathcal{A}$ that makes q oracle queries to one of the random systems, and a set Ω that contains all possible answers for oracle queries to the random systems. For a random system $\mathcal{S}$ and $i \in \{1, \ldots, q\}$, let $Z_{\mathcal{S},i}$ be the random variable over Ω that follows the distribution of the i-th answer obtained by $\mathcal{A}$ interacting with $\mathcal{S}$. Let

$$\mathbf{Z}^i_{\mathcal{S}} \stackrel{\text{def}}{=} (Z_{\mathcal{S},1}, \ldots, Z_{\mathcal{S},i}),$$

and let

$$\mathsf{p}^i_{\mathcal{S}}(\mathbf{z}) \stackrel{\text{def}}{=} \Pr\left[\mathbf{Z}^i_{\mathcal{S}} = \mathbf{z}\right]$$

for $\mathbf{z} \in \Omega^i$. For $i < q$ and $\mathbf{z} = (z_1, \ldots, z_{i-1}) \in \Omega^{i-1}$ such that $\mathsf{p}^{i-1}_{\mathcal{S}}(\mathbf{z}) > 0$, the probability distribution of $Z_{\mathcal{S},i}$ conditioned on $\mathbf{Z}^{i-1}_{\mathcal{S}} = \mathbf{z}$ will be denoted $\mathsf{p}^{\mathbf{z}}_{\mathcal{S},i}(\cdot)$,

[1] We can assume that $\mathcal{A}$ is deterministic since it is computationally unbounded.

namely for $z \in \Omega$,

$$\mathsf{p}^{\mathbf{z}}_{\mathcal{S},i}(z) \stackrel{\text{def}}{=} \Pr\left[Z_{\mathcal{S},i} = z \mid \mathbf{Z}^{i-1}_{\mathcal{S}} = \mathbf{z}\right].$$

For two random systems $\mathcal{S}_0$ and $\mathcal{S}_1$, and for $i < q$ and $\mathbf{z} = (z_1, \ldots, z_{i-1}) \in \Omega^{i-1}$ such that $\mathsf{p}^{i-1}_{\mathcal{S}_0}(\mathbf{z})$, $\mathsf{p}^{i-1}_{\mathcal{S}_1}(\mathbf{z}) > 0$, the χ^2-*divergence* for $\mathsf{p}^{\mathbf{z}}_{\mathcal{S}_0,i}(\cdot)$ and $\mathsf{p}^{\mathbf{z}}_{\mathcal{S}_1,i}(\cdot)$ is defined as follows.

$$\chi^2\left(\mathsf{p}^{\mathbf{z}}_{\mathcal{S}_1,i}(\cdot), \mathsf{p}^{\mathbf{z}}_{\mathcal{S}_0,i}(\cdot)\right) \stackrel{\text{def}}{=} \sum_{\substack{z \in \Omega \text{ such that} \\ \mathsf{p}^{\mathbf{z}}_{\mathcal{S}_0,i}(z) > 0}} \frac{\left(\mathsf{p}^{\mathbf{z}}_{\mathcal{S}_1,i}(z) - \mathsf{p}^{\mathbf{z}}_{\mathcal{S}_0,i}(z)\right)^2}{\mathsf{p}^{\mathbf{z}}_{\mathcal{S}_0,i}(z)}.$$

We will simply write $\chi^2(\mathbf{z}) = \chi^2\left(\mathsf{p}^{\mathbf{z}}_{\mathcal{S}_1,i}(\cdot), \mathsf{p}^{\mathbf{z}}_{\mathcal{S}_0,i}(\cdot)\right)$ when the random systems are clear from the context. If the support of $\mathsf{p}^{i-1}_{\mathcal{S}_1}(\cdot)$ is contained in the support of $\mathsf{p}^{i-1}_{\mathcal{S}_0}(\cdot)$, then we can view $\chi^2\left(\mathsf{p}^{\mathbf{z}}_{\mathcal{S}_1,i}(\cdot), \mathsf{p}^{\mathbf{z}}_{\mathcal{S}_0,i}(\cdot)\right)$ as a random variable, denoted $\chi^2\left(\mathbf{Z}^{i-1}_{\mathcal{S}_1}\right)$, where $\mathbf{z}$ follows the distribution of $\mathbf{Z}^{i-1}_{\mathcal{S}_1}$.

Then $\mathcal{A}$'s distinguishing advantage is upper bounded by the *total variation distance* of $\mathsf{p}^q_{\mathcal{S}_0}(\cdot)$ and $\mathsf{p}^q_{\mathcal{S}_1}(\cdot)$, denoted $\|\mathsf{p}^q_{\mathcal{S}_0}(\cdot) - \mathsf{p}^q_{\mathcal{S}_1}(\cdot)\|$, and we also have

$$\|\mathsf{p}^q_{\mathcal{S}_0}(\cdot) - \mathsf{p}^q_{\mathcal{S}_1}(\cdot)\| \leq \left(\frac{1}{2}\sum_{i=1}^{q} \mathbf{Ex}\left[\chi^2\left(\mathbf{Z}^{i-1}_{\mathcal{S}_1}\right)\right]\right)^{1/2}. \tag{1}$$

See [5] for the proof of (1).

3 Indifferentiability of TRP

We will assume that a distinguisher $\mathcal{A}$ has access to an oracle $\mathcal{O}$ with three types of queries; $\mathcal{O}(x, 0)$ for $x \in \{0,1\}^{n-\ell}$, $\mathcal{O}(u, +)$ and $\mathcal{O}(v, -)$ for $u, v \in \{0,1\}^n$, which are called a *function query*, a *forward query* and a *backward query*, respectively. Forward and backward queries will be also called *simulator queries*. In the real world, an n-bit permutation P is chosen uniformly at random, and queries $\mathcal{O}(u, +)$ and $\mathcal{O}(v, -)$ are answered with $\mathsf{P}(u)$ and $\mathsf{P}^{-1}(v)$, respectively, and a query $\mathcal{O}(x, 0)$ is answered with $\mathsf{TRP}[\mathsf{P}](x)$. In the simulated world, an $(n-\ell)$-to-$(n-m)$ bit function F is chosen uniformly at random, and a query $\mathcal{O}(x, 0)$ is answered with $\mathsf{F}(x)$ for any $x \in \{0,1\}^{n-\ell}$. On the other hand, queries $\mathcal{O}(u, +)$ and $\mathcal{O}(v, -)$ will be answered by a simulator S that has oracle access to F.

3.1 Regular Indifferentiability of TRP

We define a simulator S without using any information on the adversarial queries of type $\mathcal{O}(\cdot, 0)$. Simulator S is stateful, keeping variables $\mathcal{O}(u)$ and $\mathcal{O}^{-1}(v)$ for every u and $v \in \{0,1\}^n$, all initialized as $\perp$, meaning "undefined",[2] as well as sets $\mathcal{D}$, $\mathcal{R}$, and $\mathcal{R}_y$ for each $y \in \{0,1\}^{n-m}$, all initialized as empty. It behaves as follows.

[2] We uses $\mathcal{O}$ to denote both oracle interfaces and variables by slight abuse of notation.

– On a forward query $\mathcal{O}(u,+)$, S does the following.
  1. If $\mathcal{O}(u) = \perp$, then
     (a) obtain $y = \mathsf{F}(x)$ via an oracle query to F if $u = c \parallel x$ for some $x \in \{0,1\}^{n-\ell}$, and choose y uniformly at random from $\{0,1\}^{n-m}$ otherwise;
     (b) choose w uniformly at random from $\{0,1\}^m \setminus \mathcal{R}_y$;
     (c) assign $w \parallel y$ and u to $\mathcal{O}(u)$ and $\mathcal{O}^{-1}(w \parallel y)$, respectively;
     (d) update $\mathcal{D}$, $\mathcal{R}$ and $\mathcal{R}_y$ as $\mathcal{D}\cup\{u\}$, $\mathcal{R}\cup\{w\|y\}$ and $\mathcal{R}_y\cup\{w\}$, respectively.
  2. Return $\mathcal{O}(u)$.
– On a backward query $\mathcal{O}(v,-)$, S does the following.
  1. If $\mathcal{O}^{-1}(v) = \perp$, then
     (a) choose u uniformly at random from $\{0,1\}^n \setminus (\mathcal{D} \cup \mathcal{C})$;
     (b) assign u and v to $\mathcal{O}^{-1}(v)$ and $\mathcal{O}(u)$, respectively;
     (c) update $\mathcal{D}$, $\mathcal{R}$ and $\mathcal{R}_y$ as $\mathcal{D} \cup \{u\}$, $\mathcal{R} \cup \{v\}$ and $\mathcal{R}_y \cup \{w\}$, respectively, where $v = w \parallel y$ for $w \in \{0,1\}^m$ and $y \in \{0,1\}^{n-m}$.
  2. Return $\mathcal{O}^{-1}(v)$.

By definition, our simulator consistently answers redundant queries. So we can assume that $\mathcal{A}$ makes no redundant query; if $\mathcal{A}$ obtains $\mathcal{O}(u,+) = v$ (resp. $\mathcal{O}(v,-) = u$), then it would not make a query $\mathcal{O}(v,-)$ (resp. $\mathcal{O}(u,+)$). $\mathcal{A}$ will not make a function query $\mathsf{F}(x)$ once it has made a forward query $\mathcal{O}(c \parallel x,+)$. On the other hand, $\mathcal{A}$ is allowed to make a forward query $\mathcal{O}(c \parallel x,+)$ after it obtains $\mathsf{F}(x)$.

Theorem 1. *Let S be the simulator defined as above, and let q_F and q_S be positive integers such that $q_F + q_S \leq 2^{n-1}$. Then for any distinguisher $\mathcal{A}$ making q_F queries to the outer construction and q_S queries to the inner primitive,*

$$\boldsymbol{Adv}^{\mathsf{reg}}_{\mathsf{TRP},\mathsf{S}}(\mathcal{A}) \leq \left(\frac{(q_F+q_S)^3}{2^{n+m-1}}\right)^{\frac{1}{2}} + \frac{(3\ln q_F + 3(n-m)+1)q_S}{2^{m-1}} + \frac{5q_S}{2^{\ell-1}}.$$

Proof. We can assume that $q_S \leq 2^{m-1}$ since otherwise the upper bound trivially holds.

Let $\mathcal{S}_0 = (\mathsf{F}, \mathsf{S}[\mathsf{F}])$ and $\mathcal{S}_2 = (\mathsf{TRP}[\mathsf{P}], \mathsf{P})$ denote the simulated world and the real world, respectively. We cannot directly apply the χ^2 method to $\mathcal{S}_0$ and $\mathcal{S}_2$ since the support of $\mathsf{p}^{i-1}_{\mathcal{S}_2}(\cdot)$ is not contained in the support of $\mathsf{p}^{i-1}_{\mathcal{S}_0}(\cdot)$ (and vice versa) for any $i = 1, \ldots, q$; S does not return any element of $\mathcal{C}$ on a backward query $\mathcal{O}(\cdot,-)$. For this reason, we introduce an intermediate world, denoted $\mathcal{S}_1$, that has the same oracle interface as $\mathcal{S}_0$ and $\mathcal{S}_2$.

This random system uses two flags, denoted bad_1 and bad_2, all initialized as false, and a sampling procedure P^* as a subroutine. The procedure P^* keeps variables $\mathsf{P}^*(u)$ and $(\mathsf{P}^*)^{-1}(v)$ for every u and $v \in \{0,1\}^n$, all initialized as $\perp$, meaning "undefined", and also keeps sets $\mathcal{D}^*$ and $\mathcal{R}^*$, all initialized as empty. This procedure accepts oracle queries of types $\mathsf{P}^*(\cdot,+)$ and $\mathsf{P}^*(\cdot,-)$.

- On a query $\mathsf{P}^*(u,+)$, P^* does the following.
  1. If $\mathsf{P}^*(u) = \perp$, then
     (a) choose v uniformly at random from $\{0,1\}^n \setminus \mathcal{R}^*$;
     (b) assign v and u to $\mathsf{P}^*(u)$ and $(\mathsf{P}^*)^{-1}(v)$, respectively;
     (c) update $\mathcal{D}^*$ and $\mathcal{R}^*$ as $\mathcal{D}^* \cup \{u\}$ and $\mathcal{R}^* \cup \{v\}$, respectively.
  2. Return $\mathsf{P}^*(u)$.
- On a query $\mathsf{P}^*(v,-)$, P^* does the following.
  1. If $(\mathsf{P}^*)^{-1}(v) = \perp$, then
     (a) choose u uniformly at random from $\{0,1\}^n \setminus \mathcal{D}^*$;
     (b) if $u \in \mathcal{C}$, then set bad_1 to true, and choose u uniformly at random from $\{0,1\}^n \setminus (\mathcal{D}^* \cup \mathcal{C})$;
     (c) assign v and u to $\mathsf{P}^*(u)$ and $(\mathsf{P}^*)^{-1}(v)$, respectively;
     (d) update $\mathcal{D}^*$ and $\mathcal{R}^*$ as $\mathcal{D}^* \cup \{u\}$ and $\mathcal{R}^* \cup \{v\}$, respectively.
  2. If $(\mathsf{P}^*)^{-1}(v) = u'(\neq \perp)$ where $v = w \parallel y$ for $w \in \{0,1\}^m$ and $y \in \{0,1\}^{n-m}$, then
     (a) set bad_2 to true;
     (b) choose u uniformly at random from $\{0,1\}^n \setminus (\mathcal{D}^* \cup \mathcal{C})$;
     (c) assign v and u to $\mathsf{P}^*(u)$ and $(\mathsf{P}^*)^{-1}(v)$, respectively;
     (d) choose v' uniformly at random from
     $$\{w \parallel y : w \in \{0,1\}^m\} \setminus \mathcal{R}^*;$$
     (e) assign v' and u' to $\mathsf{P}^*(u')$ and $(\mathsf{P}^*)^{-1}(v')$, respectively;
     (f) update $\mathcal{D}^*$ and $\mathcal{R}^*$ as $\mathcal{D}^* \cup \{u\}$ and $\mathcal{R}^* \cup \{v'\}$, respectively.
  3. Return $(\mathsf{P}^*)^{-1}(v)$.

Note that $\{0,1\}^n \setminus (\mathcal{D}^* \cup \mathcal{C})$ is always nonempty since $q_F + q_S + 2^{n-\ell} \leq 2^n$. Using this sampling procedure, oracle queries to $\mathcal{S}_1$ are answered as follows.

- On a function query $\mathcal{O}(x,0)$, $\mathcal{S}_1$ obtains $w \| y = \mathsf{P}^*(c \| x, +)$ where $w \in \{0,1\}^m$ and $y \in \{0,1\}^{n-m}$, and returns y.
- On a forward query $\mathcal{O}(u,+)$, $\mathcal{S}_1$ obtains $v = \mathsf{P}^*(u,+)$ and returns v.
- On a backward query $\mathcal{O}(v,-)$, $\mathcal{S}_1$ obtains $u = \mathsf{P}^*(v,-)$ and returns u.

So $\mathcal{S}_1$ behaves like the real world $\mathcal{S}_2$ with the inner permutation replaced by the sampling procedure P^*. Again, P^* behaves like a truly random permutation except that it never samples any element of $\mathcal{C}$ on a backward query $\mathsf{P}^*(\cdot,-)$.

Note that $\mathsf{P}^*(v,-)$ is queried on an element v such that $(\mathsf{P}^*)^{-1}(v) \neq \perp$ only when $(\mathsf{P}^*)^{-1}(v)$ is fixed via a function query $\mathcal{O}(x,0)$ for some $x \in \{0,1\}^{n-\ell}$ (since we are assuming that a distinguisher never makes redundant queries). When $\mathsf{P}^*(c \parallel x) = v$ is fixed via a function query, a distinguisher would not obtain any information on the leftmost m bits of v. Namely, when $v = w \,\|\, y$ for $w \in \{0,1\}^m$ and $y \in \{0,1\}^{n-m}$, the distinguisher has $\mathsf{P}^*(u) = \star \,\|\, y$ for unknown $\star$. When a backward query $\mathsf{P}^*(v, -)$ is made later during the attack, w is replaced by a new element w' and $(\mathsf{P}^*)^{-1}(v)$ is also given a new element u' outside $\mathcal{D}^*$. In this way, every oracle query will add a new element to $\mathcal{D}^*$ and $\mathcal{R}^*$.

Let $q = q_F + q_S$ denote the total number of queries. Then we have

$$\begin{aligned}\mathbf{Adv}^{\mathsf{reg}}_{\mathsf{TRP},\mathsf{S}}(\mathcal{A}) &\le \|\mathsf{p}^q_{\mathcal{S}_0}(\cdot) - \mathsf{p}^q_{\mathcal{S}_2}(\cdot)\| \\ &\le \|\mathsf{p}^q_{\mathcal{S}_0}(\cdot) - \mathsf{p}^q_{\mathcal{S}_1}(\cdot)\| + \|\mathsf{p}^q_{\mathcal{S}_1}(\cdot) - \mathsf{p}^q_{\mathcal{S}_2}(\cdot)\|. \end{aligned} \tag{2}$$

Once $\mathcal{A}$ obtains the first $i-1$ answers $\mathbf{z} = (z_1, \ldots, z_{i-1})$ via oracle queries, they (partially) determine all the corresponding evaluations of P^*. For a fixed $j \in \{1, \ldots, i-1\}$, the j-th query is associated with (u_j, v_j, σ_j), where

- if z_j has been obtained by a function query on x, then $\sigma_j = 0$, $u_j = c \,\|\, x$, and $v_j = \star \,\|\, z_j$ (with $\star$ meaning "unknown").
- if z_j has been obtained by a forward query on u, then $\sigma_j = +$, $u_j = u$, and $v_j = z_j$.
- if z_j has been obtained by a backward query on v, then $\sigma_j = -$, $u_j = z_j$, and $v_j = v$.

With this notation, we will consider random variables V_y, S_y, F_y for each $y \in \{0,1\}^{n-m}$, where

$$\begin{aligned} V_y &= \left|\{u_j : v_j = w \,\|\, y \text{ for some } w \in \{0,1\}^m\}\right|, \\ S_y &= \left|\{u_j : \sigma_j \in \{+,-\} \text{ and } v_j = w \,\|\, y \text{ for some } w \in \{0,1\}^m\}\right|, \\ F_y &= V_y - S_y. \end{aligned}$$

In words,

- V_y counts the number of elements u where $\mathsf{P}^*(u)$ has been determined by $\mathcal{A}$'s function/simulator queries and $\mathsf{P}^*(u) = w \,\|\, y$ for some $w \in \{0,1\}^m$,
- S_y counts the number of elements u where $\mathsf{P}^*(u)$ has been determined by $\mathcal{A}$'s *simulator* queries and $\mathsf{P}^*(u) = w \,\|\, y$ for some $w \in \{0,1\}^m$,
- F_y counts the number of elements u where $\mathsf{P}^*(u)$ has been partially determined only by $\mathcal{A}$'s *function* queries and $\mathsf{P}^*(u) = \star \,\|\, y$ with *unknown* $\star \in \{0,1\}^m$.

Let $V = \sum_{y \in \{0,1\}^{n-m}} V_y$. At any point during the attack, $V = |\mathcal{D}^*| = |\mathcal{R}^*|$. Suppose that $\mathbf{z}$ determines $\mathsf{P}^*(u) = \star \| y$ for $u \in \{0,1\}^n$ and $y \in \{0,1\}^{n-m}$ (with unknown $\star$). Then for $w \in \{0,1\}^m$ such that $\mathbf{z}$ does not determine $(\mathsf{P}^*)^{-1}(w \| y)$, the conditional probability that $\star = w$ given $\mathbf{z}$ is $1/S_y$. (Note that we can define a set of candidate permutations which are compatible with (u_j, v_j, σ_j) for all $j < i$; the distribution of the next query answer $\mathbf{z}$ from $\mathcal{S}_1$ is the same as the distribution one would get by drawing one of those compatible permutations uniformly at random conditioned on backward queries not falling in $\mathcal{C}$, and using it to answer the query in the obvious way.)

Upper bounding $\|\mathsf{p}^q_{\mathcal{S}_1}(\cdot) - \mathsf{p}^q_{\mathcal{S}_2}(\cdot)\|$. The procedure P^* behaves exactly like a truly random permutation without any of the bad flags being set to true. So we can upper bound $\|\mathsf{p}^q_{\mathcal{S}_1}(\cdot) - \mathsf{p}^q_{\mathcal{S}_2}(\cdot)\|$ by the probability that either bad_1 or bad_2 is set to true.

For $i = 1, \ldots, q_S$, let $\mathsf{E}_{1,i}$ (resp. $\mathsf{E}_{2,i}$) be the event that the i-th simulator query set bad_1 (resp. bad_2) to true. Since $|\mathcal{C}| = 2^{n-\ell}$ and $|\mathcal{D}^*| \le q \le 2^{n-1}$, we have

$$\Pr[\mathsf{E}_{1,i}] = \frac{|\mathcal{C}|}{2^n - |\mathcal{D}^*|} \le \frac{2^{n-\ell}}{2^{n-1}} = \frac{1}{2^{\ell-1}}$$

for each $i = 1, \ldots, q_S$.

When the i-th simulator query $\mathcal{O}(v, -)$ is made (in the backward direction) with $v = w \,\|\, y$, the conditional probability that bad_2 is set to true (conditioned on the previous queries) is upper bounded by

$$\frac{F_y}{2^m - S_y},$$

where F_y and S_y can be viewed as random variables determined by the previous queries. Since y can be chosen adversarially and $S_y \le 2^{m-1}$, the conditional probability that the i-th simulator query sets bad_2 to true is upper bounded by

$$\frac{\max_{y \in \{0,1\}^{n-m}} F_y}{2^{m-1}}.$$

Therefore, we have

$$\Pr[\mathsf{E}_{2,i}] \le \frac{\mathbf{Ex}_i\left[\max_{y \in \{0,1\}^{n-m}} F_y\right]}{2^{m-1}},$$

where the expectation is taken over the interaction of $\mathcal{A}$ and $\mathcal{S}_1$ until the i-th simulator query is made. We also have

$$\mathbf{Ex}_i\left[\max_y F_y\right] \le \frac{q_F}{2^{n-m-2}} + 3\ln q_F + 3(n-m) + 1. \tag{3}$$

The proof of (3) is deferred to the end of this section. Overall, we have

$$\begin{aligned}
\|\mathsf{p}^q_{\mathcal{S}_1}(\cdot) - \mathsf{p}^q_{\mathcal{S}_2}(\cdot)\| &\le \Pr\left[\bigvee_{i=1}^{q_S} (\mathsf{E}_{1,i} \vee \mathsf{E}_{2,i})\right] \\
&\le \sum_{i=1}^{q_S} \Pr[\mathsf{E}_{1,i}] + \sum_{i=1}^{q_S} \Pr[\mathsf{E}_{2,i}] \\
&\le \frac{q_S}{2^{\ell-1}} + \frac{q_F q_S}{2^{n-3}} + \frac{(3\ln q_F + 3(n-m) + 1)q_S}{2^{m-1}} \\
&\le \frac{5q_S}{2^{\ell-1}} + \frac{(3\ln q_F + 3(n-m) + 1)q_S}{2^{m-1}}, \qquad (4)
\end{aligned}$$

where the last inequality holds since $q_F \leq 2^{n-\ell}$.

UPPER BOUNDING $\|\mathsf{p}^q_{\mathcal{S}_0}(\cdot) - \mathsf{p}^q_{\mathcal{S}_1}(\cdot)\|$. For the intermediate system $\mathcal{S}_1$, we can easily check that the support of $\mathsf{p}^{i-1}_{\mathcal{S}_1}(\cdot)$ is contained in the support of $\mathsf{p}^{i-1}_{\mathcal{S}_0}(\cdot)$ for $i = 1, \ldots, q$, allowing us to use the χ^2 method.

Let $\Omega = \{0,1\}^n \cup \{0,1\}^{n-m}$. For fixed $i \in \{1, \ldots, q\}$ and $\mathbf{z} \in \Omega^{i-1}$ such $\mathsf{p}^{i-1}_{\mathcal{S}_1}(\mathbf{z}) > 0$, we will compute

$$\chi^2(\mathbf{z}) = \sum_{\substack{z \in \Omega \text{ such that} \\ \mathsf{p}^{\mathbf{z}}_{\mathcal{S}_0,i}(z) > 0}} \frac{\left(\mathsf{p}^{\mathbf{z}}_{\mathcal{S}_1,i}(z) - \mathsf{p}^{\mathbf{z}}_{\mathcal{S}_0,i}(z)\right)^2}{\mathsf{p}^{\mathbf{z}}_{\mathcal{S}_0,i}(z)}.$$

The previous queries $\mathbf{z} \in \Omega^{i-1}$ will determine the type of the next query. We will distinguish four cases: a function query, a "fresh" forward query, a forward query on an element where a function query already has been made, and a backward query.

Suppose that the i-th query is a function query. For any $z \in \{0,1\}^{n-m}$, we have

$$\mathsf{p}^{\mathbf{z}}_{\mathcal{S}_0,i}(z) = \frac{1}{2^{n-m}},$$
$$\mathsf{p}^{\mathbf{z}}_{\mathcal{S}_1,i}(z) = \frac{2^m - V_z}{2^n - V}$$

since $V = |\mathcal{R}^*|$ and $V_z = |\{v \in \mathcal{R}^* : v = w \,\|\, z \text{ for some } w \in \{0,1\}^m\}|$. Therefore we have

$$\chi^2(\mathbf{z}) = \sum_{z \in \{0,1\}^{n-m}} \frac{(2^{n-m}V_z - V)^2}{2^{n-m}(2^n - V)^2}. \tag{5}$$

Suppose that the i-th query is a forward query $\mathcal{O}(u,+)$, where either $u \notin \mathcal{C}$ or $u = c \,\|\, x$ for some $x \in \{0,1\}^{n-\ell}$ and $\mathcal{O}(x,0)$ has not been queried. Let $z = w \,\|\, y$ for $w \in \{0,1\}^m$ and $y \in \{0,1\}^{n-m}$, where $(P^*)^{-1}(w \,\|\, y)$ is not fixed by $\mathbf{z}$. Then it is easy to see that

$$\mathsf{p}^{\mathbf{z}}_{\mathcal{S}_0,i}(z) = \frac{1}{2^{n-m}} \cdot \frac{1}{2^m - S_y}.$$

In $\mathcal{S}_1$, $\perp \| y$ is chosen with probability $(2^m - V_y)/(2^n - V)$ conditioned on $\mathbf{z}$ (with $\perp$ meaning "undetermined"), and then $\perp$ becomes w with probability $1/(2^m - S_y)$. Therefore we have

$$\mathsf{p}^{\mathbf{z}}_{\mathcal{S}_1,i}(z) = \frac{2^m - V_y}{2^n - V} \cdot \frac{1}{2^m - S_y},$$

and hence,

$$\chi^2(\mathbf{z}) = \sum_{y \in \{0,1\}^{n-m}} \frac{(2^{n-m}V_y - V)^2}{2^{n-m}(2^n - V)^2}, \tag{6}$$

since the number of $w \in \{0,1\}^m$ such that $(P^*)^{-1}(w \,\|\, y)$ is fixed by $\mathbf{z}$ is S_y for each $y \in \{0,1\}^m$.

Suppose that the i-th query is a forward query $\mathcal{O}(u,+)$, where $u = c \parallel x$ for some $x \in \{0,1\}^{n-\ell}$ and $y = \mathcal{O}(x,0)$ has been obtained by a previous function query. Let $z = w \parallel y$ where $w \in \{0,1\}^m$. Then we have

$$\mathsf{p}^{\mathbf{z}}_{\mathcal{S}_0,i}(z) = \mathsf{p}^{\mathbf{z}}_{\mathcal{S}_1,i}(z) = \frac{1}{2^m - S_y},$$

and hence

$$\chi^2(\mathbf{z}) = 0. \tag{7}$$

Suppose that the i-th query is a backward query $\mathcal{O}(v,-)$. It is easy to see that

$$\mathsf{p}^{\mathbf{z}}_{\mathcal{S}_0,i}(z) = \mathsf{p}^{\mathbf{z}}_{\mathcal{S}_1,i}(z) = \frac{1}{2^n - |\mathcal{D}^* \cup \mathcal{C}|}$$

for any $z \in \{0,1\}^n \setminus (\mathcal{D}^* \cup \mathcal{C})$, and hence

$$\chi^2(\mathbf{z}) = 0. \tag{8}$$

By (5), (6), (7), (8), we have

$$\begin{aligned}
\|\mathsf{p}^q_{\mathcal{S}_0}(\cdot) - \mathsf{p}^q_{\mathcal{S}_1}(\cdot)\| &\le \left(\frac{1}{2}\sum_{i=1}^{q} \mathbf{Ex}\left[\chi^2(\mathbf{z})\right]\right)^{\frac{1}{2}} \\
&\le \left(\frac{1}{2}\sum_{i=1}^{q} \mathbf{Ex}\left[\sum_{y\in\{0,1\}^{n-m}} \frac{(2^{n-m}V_y - V)^2}{2^{n-m}(2^n - V)^2}\right]\right)^{\frac{1}{2}}.
\end{aligned} \tag{9}$$

Since $\sum_{y\in\{0,1\}^{n-m}} V_y = V \le q_F + q_S$ and $V \le 2^{n-1}$, we have

$$\begin{aligned}
\sum_{y\in\{0,1\}^{n-m}} \frac{(2^{n-m}V_y - V)^2}{2^{n-m}(2^n - V)^2} &= \sum_{y\in\{0,1\}^{n-m}} \frac{2^{2n-2m}V_y^2 - 2^{n-m+1}V_yV + V^2}{2^{n-m}(2^n - V)^2} \\
&= \frac{2^{n-m}}{(2^n - V)^2}\left(\sum_{y\in\{0,1\}^{n-m}} V_y^2 - \frac{V^2}{2^{n-m}}\right) \\
&\le \frac{1}{2^{n+m-2}}\left(\sum_{y\in\{0,1\}^{n-m}} V_y\right)^2 \\
&\le \frac{(q_F + q_S)^2}{2^{n+m-2}},
\end{aligned}$$

and by (9),

$$\|\mathsf{p}^q_{\mathcal{S}_0}(\cdot) - \mathsf{p}^q_{\mathcal{S}_1}(\cdot)\| \le \left(\sum_{i=1}^{q} \frac{(q_F + q_S)^2}{2^{n+m-1}}\right)^{\frac{1}{2}} = \left(\frac{(q_F + q_S)^3}{2^{n+m-1}}\right)^{\frac{1}{2}}. \tag{10}$$

By (2), (4), (10), the proof is complete. $\square$

When $q_S = 0$, we can obtain a tighter upper bound on $\|\mathsf{p}^q_{\mathcal{S}_0}(\cdot) - \mathsf{p}^q_{\mathcal{S}_1}(\cdot)\|$ than the one obtained above, recovering the optimal indistinguishability bound of TRP given in [8]. See Appendix A.

Proof of (3). For any function query $\mathcal{O}(x, 0)$ and for any $y \in \{0,1\}^{n-m}$, the probability that $\mathcal{O}(x, 0) = y$ is upper bounded by

$$\frac{2^m}{2^n - (q_F + q_S)} \le \frac{1}{2^{n-m-1}}.$$

Let X be a random variable that follows the binomial distribution with parameters q_F and $p = 1/2^{n-m-1}$, namely,

$$\Pr[X = j] = \binom{q_F}{j} p^j (1-p)^{q_F - j}$$

for $j = 0, \ldots, q_F$. Then for any $y \in \{0,1\}^{n-m}$, we have

$$\Pr[F_y \ge j] \le \Pr[X \ge j].$$

By the Chernoff bound, we have

$$\Pr[X \ge j] \le e^{-\frac{j - pq_F}{3}} \le \frac{p}{2q_F}$$

for any $j \ge 2pq_F + 3 \ln \frac{2q_F}{p}$. Therefore we have

$$\begin{aligned}
\mathbf{Ex}\left[\max_y F_y\right] &= \sum_{j \ge 1} \Pr\left[\max_y F_y \ge j\right] \\
&\le 2pq_F + 3 \ln \frac{2q_F}{p} + \sum_{j > 2pq_F + 3 \ln \frac{2q_F}{p}} \Pr\left[\max_y F_y \ge j\right] \\
&= 2pq_F + 3 \ln \frac{2q_F}{p} + \sum_{j > 2pq_F + 3 \ln \frac{2q_F}{p}} \Pr\left[\bigvee_{y \in \{0,1\}^{n-m}} F_y \ge j\right] \\
&\le 2pq_F + 3 \ln \frac{2q_F}{p} + \sum_{y \in \{0,1\}^{n-m}} \sum_{j > 2pq_F + 3 \ln \frac{2q_F}{p}} \Pr[X \ge j] \\
&\le 2pq_F + 3 \ln \frac{2q_F}{p} + 2^{n-m} \cdot q_F \cdot \frac{p}{2q_F} \\
&\le \frac{q_F}{2^{n-m-2}} + 3 \ln q_F + 3(n-m) + 1.
\end{aligned}$$

3.2 Public Indifferentiability of TRP

We define a simulator S which is stateful, keeping variables $\mathcal{O}(u)$ and $\mathcal{O}^{-1}(v)$ for every u and $v \in \{0,1\}^n$, all initialized as $\perp$, meaning "undefined", as well as sets $\mathcal{D}$, $\mathcal{R}$, and $\mathcal{R}_y$ for each $y \in \{0,1\}^{n-m}$, all initialized as empty. It also uses a special symbol $\circledast$ (not in $\{0,1\}^{n-m}$). We will call oracle queries $\mathcal{O}(u, +)$ (resp. $\mathcal{O}(v, -)$) *fresh* if $\mathcal{O}(u) = \perp$ (resp. $\mathcal{O}^{-1}(v) = \perp$).

- On a fresh forward query $\mathcal{O}(u,+)$, S does the following.
  1. If $u = c \,\|\, x$ for some $x \in \{0,1\}^{n-\ell}$ (i.e., $u \in \mathcal{C}$), then obtain $y = \mathsf{F}(x)$ via an oracle query to F.
     (a) If $\mathcal{R}_y \neq \{0,1\}^m$, then
        i. choose w uniformly at random from $\{0,1\}^m \setminus \mathcal{R}_y$;
        ii. assign $w \,\|\, y$ and u to $\mathcal{O}(u)$ and $\mathcal{O}^{-1}(w \,\|\, y)$, respectively;
        iii. update $\mathcal{D}$, $\mathcal{R}$ and $\mathcal{R}_y$ as $\mathcal{D} \cup \{u\}$, $\mathcal{R} \cup \{w \,\|\, y\}$ and $\mathcal{R}_y \cup \{w\}$, respectively;
        iv. return $\mathcal{O}(u)$.
     (b) If $\mathcal{R}_y = \{0,1\}^m$, then return $\circledast \,\|\, y$.
  2. If $u \notin \mathcal{C}$, then
     (a) choose v uniformly at random from $\{0,1\}^n \setminus \mathcal{R}$;
     (b) assign v and u to $\mathcal{O}(u)$ and $\mathcal{O}^{-1}(v)$, respectively;
     (c) update $\mathcal{D}$, $\mathcal{R}$ and $\mathcal{R}_y$ as $\mathcal{D} \cup \{u\}$, $\mathcal{R} \cup \{v\}$ and $\mathcal{R}_y \cup \{w\}$, respectively, where $v = w \,\|\, y$ for $w \in \{0,1\}^m$ and $y \in \{0,1\}^{n-m}$;
     (d) return $\mathcal{O}(u)$.
- On a fresh backward query $\mathcal{O}(v,-)$, S does the following.
  1. Choose u uniformly at random from $\{0,1\}^n \setminus (\mathcal{D} \cup \mathcal{C})$.
  2. Assign v and u to $\mathcal{O}(u)$ and $\mathcal{O}^{-1}(v)$, respectively.
  3. Update $\mathcal{D}$, $\mathcal{R}$ and $\mathcal{R}_y$ as $\mathcal{D} \cup \{u\}$, $\mathcal{R} \cup \{v\}$ and $\mathcal{R}_y \cup \{w\}$, respectively, where $v = w \,\|\, y$ for $w \in \{0,1\}^m$ and $y \in \{0,1\}^{n-m}$.
  4. Return $\mathcal{O}^{-1}(v)$.
- On a forward query $\mathcal{O}(u,+)$ (resp. a backward query $\mathcal{O}(v,-)$) which is not fresh, S returns $\mathcal{O}(u)$ (resp. $\mathcal{O}^{-1}(v)$).

In the public indifferentiability model, the simulator knows all queries made by the distinguisher to F. When a distinguisher makes a function query $\mathcal{O}(x,0)$, S will behave exactly in the same manner as it would have done with a forward query $\mathcal{O}(c \,\|\, x,+)$, except returning the response.

Theorem 2. *Let S be the simulator defined as above, and let q_F and q_S be positive integers such that $q_F + q_S \leq 2^{n-1}$. Then for any distinguisher $\mathcal{A}$ making q_F queries to the outer construction and q_S queries to the inner primitive,*

$$\boldsymbol{Adv}^{\mathsf{pub}}_{\mathsf{TRP},\mathsf{S}}(\mathcal{A}) \leq \begin{cases} \left(\frac{(q_F+q_S)^3}{2^{n+m-1}}\right)^{\frac{1}{2}} + \frac{q_S}{2^{\ell-1}} & \text{if } q_F + q_S < 2^m, \\ \left(\frac{5(q_F+q_S)^2}{2^{n+1}}\right)^{\frac{1}{2}} + \frac{q_S}{2^{\ell-1}} & \text{otherwise.} \end{cases}$$

Proof. By the definition of the simulator, we can assume that $\mathcal{A}$ makes a forward query $\mathcal{O}(c \,\|\, x,+)$ and then truncates the leftmost m bits (or $\circledast$) of the response when it wants to obtain $\mathcal{O}(x,0)$; this modification would not degrade the adversarial distinguishing advantage. So we can remove the oracle interface $\mathcal{O}(\cdot,0)$ in both the simulated world and the real world. Instead, the number of forward queries and backward queries should be upper bounded by $q_F + q_S$ and q_S, respectively. We can still assume that $\mathcal{A}$ does not make redundant queries.

Let $\mathcal{S}_0 = \mathsf{S}[\mathsf{F}]$ and $\mathcal{S}_2 = \mathsf{P}$ denote the simulated world and the real world, respectively. As in the regular indifferentiability proof, we introduce an intermediate world, denoted $\mathcal{S}_1$, that has the same oracle interface as $\mathcal{S}_0$ and $\mathcal{S}_2$. This random system uses a flag, denoted bad and initialized as false, and keeps sets $\mathcal{D}$ and $\mathcal{R}$, all initialized as empty. Oracle queries to $\mathcal{S}_1$ are answered as follows.

- On a forward query $\mathcal{O}(u,+)$, $\mathcal{S}_1$ does the following.
  1. Choose v uniformly at random from $\{0,1\}^n \setminus \mathcal{R}$.
  2. Update $\mathcal{D}$ and $\mathcal{R}$ as $\mathcal{D} \cup \{u\}$ and $\mathcal{R} \cup \{v\}$, respectively.
  3. Return v.
- On a backward query $\mathcal{O}(v,-)$, $\mathcal{S}_1$ does the following.
  1. Choose u uniformly at random from $\{0,1\}^n \setminus \mathcal{D}$.
  2. if $u \in \mathcal{C}$, then set bad to true, and choose u uniformly at random from $\{0,1\}^n \setminus (\mathcal{D} \cup \mathcal{C})$.
  3. Update $\mathcal{D}$ and $\mathcal{R}$ as $\mathcal{D} \cup \{u\}$ and $\mathcal{R} \cup \{v\}$, respectively.
  4. Return u.

So $\mathcal{S}_1$ behaves like a truly random permutation except that it does not sample any element of $\mathcal{C}$ on a backward query $\mathcal{O}(\cdot,-)$. Let $q = q_F + q_S$ denote the total number of queries. Then we have

$$\begin{aligned}\mathbf{Adv}^{\mathsf{pub}}_{\mathsf{TRP},\mathsf{S}}(\mathcal{A}) &\le \|\mathsf{p}^q_{\mathcal{S}_0}(\cdot) - \mathsf{p}^q_{\mathcal{S}_2}(\cdot)\| \\ &\le \|\mathsf{p}^q_{\mathcal{S}_0}(\cdot) - \mathsf{p}^q_{\mathcal{S}_1}(\cdot)\| + \|\mathsf{p}^q_{\mathcal{S}_1}(\cdot) - \mathsf{p}^q_{\mathcal{S}_2}(\cdot)\|. \qquad (11)\end{aligned}$$

We will consider a random variable V_y for each $y \in \{0,1\}^{n-m}$, where

$$V_y = |\{v \in \{0,1\}^n : v = w \,\|\, y \in \mathcal{R} \text{ for some } w \in \{0,1\}^m\}|\,.$$

We also define random variables

$$\begin{aligned}V &= \sum_{y\in\{0,1\}^{n-m}} V_y,\\ H &= |\{y : V_y = 2^m\}|\,.\end{aligned}$$

It is easy to see that $V = |\mathcal{D}| = |\mathcal{R}|$ at any point during the attack.

Upper bounding $\|\mathsf{p}^q_{\mathcal{S}_1}(\cdot) - \mathsf{p}^q_{\mathcal{S}_2}(\cdot)\|$. The system $\mathcal{S}_1$ behaves exactly like the real world $\mathcal{S}_2$ without the bad flag bad being set to true. So we can upper bound $\|\mathsf{p}^q_{\mathcal{S}_1}(\cdot) - \mathsf{p}^q_{\mathcal{S}_2}(\cdot)\|$ 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 backward query sets bad to true. Since $|\mathcal{C}| = 2^{n-\ell}$ and $|\mathcal{D}| \le q \le 2^{n-1}$, we have

$$\Pr[\mathsf{E}_i] = \frac{|\mathcal{C}|}{2^n - |\mathcal{D}|} \le \frac{2^{n-\ell}}{2^{n-1}} = \frac{1}{2^{\ell-1}}$$

for each $i = 1, \ldots, q_S$. Therefore, we have

$$\|\mathsf{p}^q_{\mathcal{S}_1}(\cdot) - \mathsf{p}^q_{\mathcal{S}_2}(\cdot)\| \le \Pr\left[\bigvee_{i=1}^{q_S} \mathsf{E}_i\right] \le \sum_{i=1}^{q_S} \Pr[\mathsf{E}_i] \le \frac{q_S}{2^{\ell-1}}. \tag{12}$$

Upper bounding $\|\mathsf{p}^q_{\mathcal{S}_0}(\cdot) - \mathsf{p}^q_{\mathcal{S}_1}(\cdot)\|$. For the intermediate system $\mathcal{S}_1$, we can easily check that the support of $\mathsf{p}^{i-1}_{\mathcal{S}_1}(\cdot)$ is contained in the support of $\mathsf{p}^{i-1}_{\mathcal{S}_0}(\cdot)$ for $i = 1, \ldots, q$, allowing us to use the χ^2 method. Any element of $\{\circledast\} \times \{0,1\}^{n-m}$ is returned only in $\mathbf{S}_0$.

Let $\Omega = \{0,1\}^n \cup (\{\circledast\} \times \{0,1\}^{n-m})$. For fixed $i \in \{1, \ldots, q\}$ and $\mathbf{z} \in \Omega^{i-1}$ such $\mathsf{p}^{i-1}_{\mathcal{S}_1}(\mathbf{z}) > 0$, we will compute

$$\chi^2(\mathbf{z}) = \sum_{\substack{z \in \Omega \text{ such that} \\ \mathsf{p}^{\mathbf{z}}_{\mathcal{S}_0,i}(z) > 0}} \frac{\left(\mathsf{p}^{\mathbf{z}}_{\mathcal{S}_1,i}(z) - \mathsf{p}^{\mathbf{z}}_{\mathcal{S}_0,i}(z)\right)^2}{\mathsf{p}^{\mathbf{z}}_{\mathcal{S}_0,i}(z)}.$$

The previous queries $\mathbf{z} \in \Omega^{i-1}$ determine random variables H, $V (= i-1)$ as well as the type of the next query. We will distinguish three cases: a forward query $\mathcal{O}(u, +)$ for $u \in \mathcal{C}$, a forward query $\mathcal{O}(u, +)$ for $u \notin \mathcal{C}$, and a backward query $\mathcal{O}(v, -)$.

Suppose that the i-th query is a forward query $\mathcal{O}(u, +)$, where $u \in \mathcal{C}$. If $z = \circledast \,\|\, y$ for $y \in \{0,1\}^{n-m}$ such that $|\mathcal{R}_y| = 2^{n-m}$, then

$$\mathsf{p}^{\mathbf{z}}_{\mathcal{S}_0,i}(z) = \frac{1}{2^{n-m}},$$
$$\mathsf{p}^{\mathbf{z}}_{\mathcal{S}_1,i}(z) = 0.$$

If $z \in \{0,1\}^n \setminus \mathcal{R}$, then

$$\mathsf{p}^{\mathbf{z}}_{\mathcal{S}_0,i}(z) = \frac{1}{2^{n-m}} \cdot \frac{1}{2^m - V_y},$$
$$\mathsf{p}^{\mathbf{z}}_{\mathcal{S}_1,i}(z) = \frac{1}{2^n - V}.$$

Since the number of elements $y \in \{0,1\}^{n-m}$ such that $|\mathcal{R}_y| = 2^{n-m}$ is H, we have

$$\chi^2(\mathbf{z}) = \frac{H}{2^{n-m}} + \sum_{z \in \{0,1\}^n \setminus \mathcal{R}} \frac{(2^{n-m}V_y - V)^2}{(2^n - V)^2(2^n - 2^{n-m}V_y)}. \tag{13}$$

For each $y \in \{0,1\}^{n-m}$, the number of elements $w \in \{0,1\}^m$ such that $w \parallel y \in \{0,1\}^n \setminus \mathcal{R}$ is $2^m - V_y$. Furthermore, $\sum_{y\in\{0,1\}^{n-m}} V_y = V$ and $V_y \le 2^m$ for every $y \in \{0,1\}^{n-m}$. Therefore we have

$$\begin{aligned}
\sum_{z\in\{0,1\}^n\setminus\mathcal{R}} \frac{(2^{n-m}V_y - V)^2}{(2^n - V)^2(2^n - 2^{n-m}V_y)} &= \sum_{y\in\{0,1\}^{n-m}} \frac{(2^{n-m}V_y - V)^2}{2^{n-m}(2^n - V)^2} \\
&= \frac{2^{n-m}}{(2^n - V)^2}\left(\sum_{y\in\{0,1\}^{n-m}} V_y^2 - \frac{V^2}{2^{n-m}}\right) \\
&\le \frac{1}{2^{n+m-2}} \sum_{y\in\{0,1\}^{n-m}} V_y^2 \\
&\le \frac{\min\{V^2, 2^m V\}}{2^{n+m-2}} \\
&\le \frac{\min\{q^2, 2^m q\}}{2^{n+m-2}}. \qquad (14)
\end{aligned}$$

Since $H \le \lfloor \frac{V}{2^m} \rfloor$ and $V \le q$, we have $\frac{H}{2^{n-m}} = 0$ if $q < 2^m$, and $\frac{H}{2^{n-m}} \le \frac{q}{2^n}$ otherwise. By (13) and (14), we conclude that

$$\chi^2(\mathbf{z}) \le \begin{cases} \frac{q^2}{2^{n+m-2}} & \text{if } q < 2^m, \\ \frac{5q}{2^n} & \text{otherwise.} \end{cases} \qquad (15)$$

Suppose that the i-th query is a forward query $\mathcal{O}(u, +)$, where $u \notin \mathcal{C}$. For any $z \in \{0,1\}^n \setminus \mathcal{R}$ we have

$$\mathsf{p}^{\mathbf{z}}_{\mathcal{S}_0,i}(z) = \mathsf{p}^{\mathbf{z}}_{\mathcal{S}_1,i}(z) = \frac{1}{2^n - V},$$

and hence

$$\chi^2(\mathbf{z}) = 0. \qquad (16)$$

Suppose that the i-th query is a backward query $\mathcal{O}(v, -)$. For any $z \in \{0,1\}^n \setminus (\mathcal{D} \cup \mathcal{C})$ we have

$$\mathsf{p}^{\mathbf{z}}_{\mathcal{S}_0,i}(z) = \mathsf{p}^{\mathbf{z}}_{\mathcal{S}_1,i}(z) = \frac{1}{2^n - |\mathcal{D} \cup \mathcal{C}|},$$

and hence

$$\chi^2(\mathbf{z}) = 0. \qquad (17)$$

By (15), (16), (17), we have

$$\begin{aligned}
\|\mathsf{p}^q_{\mathcal{S}_0}(\cdot) - \mathsf{p}^q_{\mathcal{S}_1}(\cdot)\| &\le \left(\frac{1}{2}\sum_{i=1}^{q} \mathbf{Ex}\left[\chi^2(\mathbf{z})\right]\right)^{\frac{1}{2}} \\
&\le \begin{cases} \left(\frac{q^3}{2^{n+m-1}}\right)^{\frac{1}{2}} & \text{if } q < 2^m, \\ \left(\frac{5q^2}{2^{n+1}}\right)^{\frac{1}{2}} & \text{otherwise.} \end{cases} \qquad (18)
\end{aligned}$$

By (11), (12), (18), the proof is complete. □

4 Tightness of Regular Indifferentiability

We can prove that our regular indifferentiability bound is tight with respect to the total number of queries $q = q_F + q_S$ when $m + \ell \ll n$. Note that if $m+\ell \ll n$ then $\min\{m, \ell\} \leq \frac{n+m}{3}$. We will assume that the number of F-queries that a simulator makes for each query of the distinguisher is a polynomial in n, denoted $\mathtt{poly}(n)$.

First, suppose that $m \leq \ell$. In this case, we consider a distinguisher $\mathcal{A}$ that begins the attack by obtaining $y = \mathsf{F}(x)$ for a random element x via a function query to F. Then $\mathcal{A}$ makes 2^m backward queries at $w \,\|\, y$, where $w \in \{0,1\}^m$. With high probability, $\mathcal{A}$ should be able to obtain $c \,\|\, x$ for some $x \in \{0,1\}^{n-\ell}$ as a response if the simulator faithfully reproduces $(\mathsf{TRP}[\mathsf{P}], \mathsf{P})$. Furthermore, it should be the case that $\mathsf{F}(x) = y$, while it is infeasible for the simulator to find a preimage of y under F (without any information of the adversarial function query) using at most 2^m queries to F if $\mathtt{poly}(n) \cdot 2^m \ll 2^{n-\ell}$. So we conclude that if $m+\ell \ll n$ then there is no simulator which is secure against any distinguisher that makes about 2^m simulator queries.

Next, suppose that $\ell \leq m$. In this attack, a distinguisher $\mathcal{A}$ randomly chooses an element $y \in \{0,1\}^{n-m}$, and makes 2^ℓ backward queries at $w \,\|\, y$, where $w \in \{0,1\}^m$. With high probability, $\mathcal{A}$ will obtain $c \,\|\, x$ for some $x \in \{0,1\}^{n-\ell}$ as a response if the simulator behaves like a random permutation. Furthermore, it should be the case that $\mathsf{F}(x) = y$. In this way, $\mathcal{A}$ is able to find a preimage of y under F using at most 2^ℓ queries to F, which is infeasible if $\mathtt{poly}(n) \cdot 2^\ell \ll 2^{n-m}$. So we conclude that if $m + \ell \ll n$ then there is no simulator which is secure against any distinguisher that makes about 2^ℓ simulator queries. Note that the second attack holds even in the public indifferentiability setting.

References

[1] M. Bellare and R. Impagliazzo. A Tool for Obtaining Tighter Security Analyses of Pseudorandom Function Based Constructions, with Applications to PRP to PRF Conversion. In *IACR Cryptology ePrint Archive 1999/024*, 1999.

[2] M. Bellare, T. Krovetz, and P. Rogaway. Luby-Rackoff backwards: Increasing security by making block ciphers non-invertible. In K. Nyberg, editor, *EUROCRYPT'98*, pages 266–280, 1998.

[3] S. Bhattacharya and M. Nandi. Full Indifferentiable Security of the Xor of Two or More Random Permutations Using the χ^2 Method. In J. B. Nielsen and V. Rijmen, editors, *EUROCRYPT 2018*, pages 387–412, 2018.

[4] B. Cogliati, R. Lampe, and J. Patarin. The Indistinguishability of the XOR of k Permutations. In C. Cid and C. Rechberger, editors, *FSE 2014*, pages 285–302, 2015.

[5] W. Dai, V. T. Hoang, and S. Tessaro. Information-Theoretic Indistinguishability via the Chi-Squared Method. In J. Katz and H. Shacham, editors, *CRYPTO 2017*, pages 497–523, 2017.

[6] Y. Dodis, L. Reyzin, R. L. Rivest, and E. Shen. Indifferentiability of Permutation-Based Compression Functions and Tree-Based Modes of Operation, with Applications to MD6. In O. Dunkelman, editor, *FSE 2009*, pages 104–121, 2009.

[7] Y. Dodis, T. Ristenpart, and T. Shrimpton. Salvaging Merkle-Damgård for practical applications. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 371–388, 2009.

[8] S. Gilboa, S. Gueron, and B. Morris. How Many Queries are Needed to Distinguish a Truncated Random Permutation from a Random Function? *Journal of Cryptology*, 31(1):162–171, Jan 2018.

[9] S. Gueron, A. Langley, and Y. Lindell. AES-GCM-SIV: Specification and Analysis. *IACR Cryptology ePrint Archive*, 2017:168, 2017.

[10] S. Gueron and Y. Lindell. GCM-SIV: Full nonce misuse-resistant authenticated encryption at under one cycle per byte. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 109–119, 2015.

[11] C. Hall, D. Wagner, J. Kelsey, and B. Schneier. Building PRFs from PRPs. In H. Krawczyk, editor, *CRYPTO'98*, pages 370–389, 1998.

[12] T. Iwata and Y. Seurin. Reconsidering the security bound of AES-GCM-SIV. *IACR Transactions on Symmetric Cryptology*, pages 240–267, 2017.

[13] J. Lee. Indifferentiability of the Sum of Random Permutations Toward Optimal Security. *IEEE Transactions on Information Theory*, 63(6):4050–4054, 2017.

[14] S. Lucks. The Sum of PRPs Is a Secure PRF. In B. Preneel, editor, *EUROCRYPT 2000*, pages 470–484, 2000.

[15] A. Mandal, J. Patarin, and V. Nachef. Indifferentiability beyond the Birthday Bound for the Xor of Two Public Random Permutations. In G. Gong and K. C. Gupta, editors, *INDOCRYPT 2010*, pages 69–81, 2010.

[16] A. Mandal, J. Patarin, and Y. Seurin. On the Public Indifferentiability and Correlation Intractability of the 6-Round Feistel Construction. In R. Cramer, editor, *TCC 2012*, pages 285–302, 2012.

[17] U. Maurer, R. Renner, and C. Holenstein. Indifferentiability, Impossibility Results on Reductions, and Applications to the Random Oracle Methodology. In M. Naor, editor, *TCC 2004*, pages 21–39, 2004.

[18] B. Mennink. Linking Stam's Bounds with Generalized Truncation. In M. Matsui, editor, *CT-RSA 2019*, pages 313–329, 2019.

[19] B. Mennink and B. Preneel. On the XOR of Multiple Random Permutations. In T. Malkin, V. Kolesnikov, A. B. Lewko, and M. Polychronakis, editors, *ACNS 2015*, pages 619–634, 2015.

[20] J. Patarin. A Proof of Security in $O(2^n)$ for the Xor of Two Random Permutations. In R. Safavi-Naini, editor, *ICITS 2008*, pages 232–248, 2008.

[21] T. Shrimpton and M. Stam. Building a collision-resistant compression function from non-compressing primitives. In *International Colloquium on Automata, Languages, and Programming*, pages 643–654, 2008.

[22] K. Yoneyama, S. Miyagawa, and K. Ohta. Leaky Random Oracle. In J. Baek, F. Bao, K. Chen, and X. Lai, editors, *ProvSec 2008*, pages 226–240, 2008.

A Indistinguishability of TRP

A *hypergeometric random distribution* $\mathsf{HG}_{N,M,q}$, parameterized by N, M, and q, is a probability distribution that describes the probability that exactly k elements are selected from a subset of M "good" elements when q elements are selected from the universe of N elements without replacement; this probability is precisely $\binom{M}{k}\binom{N-M}{n-k}/\binom{N}{n}$.

If a distinguisher makes no simulator query (namely, $q_S = 0$) when it interacts with $\mathcal{S}_1$ in the regular indifferentiability setting, then V_y would follow the hypergeometric distribution with $N = 2^n$, $M = 2^m$ and $q = i - 1 (= V)$. In this case, it is well known that

$$\begin{aligned}\mathbf{Ex}[V_y] &= \frac{V}{2^{n-m}},\\ \mathbf{Var}[V_y] &= \frac{2^m(2^n - 2^m)(2^n - V)V}{2^{2n}(2^n - 1)}.\end{aligned}$$

Since

$$\mathbf{Var}[V_y] = \mathbf{Ex}[V_y^2] - \mathbf{Ex}[V_y]^2,$$

and

$$\begin{aligned}\sum_{y \in \{0,1\}^{n-m}} \mathbf{Var}[V_y] &\leq 2^{n-m}\left(\frac{2^m(2^n - 2^m)(2^n - V)V}{2^{2n}(2^n - 1)}\right)\\ &\leq \frac{2^m(2^n - 2^m)V}{2^{n+m}}\\ &\leq V \leq q_F,\end{aligned}$$

we have

$$\begin{aligned}\mathbf{Ex}\left[\sum_y \frac{(2^{n-m}V_y - V)^2}{2^{n-m}(2^n - V)^2}\right] &\leq \frac{1}{2^{n+m-2}} \sum_{y \in \{0,1\}^{n-m}} \left(\mathbf{Ex}\left[V_y^2\right] - \mathbf{Ex}\left[V_y\right]^2\right)\\ &\leq \frac{q_F}{2^{n+m-2}}.\end{aligned}$$

Plugging this into (9), we obtain the indistinguishability bound of TRP as follows.

$$\mathbf{Adv}^{\mathsf{ind}}_{\mathsf{TRP}}(\mathcal{A}) \leq \frac{q}{2^{\frac{n+m-1}{2}}},$$

for any distinguisher $\mathcal{A}$ making q queries.