

Indifferentiability Beyond the Birthday Bound for the Xor of two Public Random Permutations

Avradip Mandal¹, Jacques Patarin², and Valerie Nachev³

¹ University of Luxembourg, Luxembourg
avradip.mandal@uni.lu

² PRISM, Université de Versailles, France
jacques.patarin@prism.uvsq.fr

³ UMR CNRS 8088, University of Cergy-Pontoise, France
valerie.nachev@u-cergy.fr

Abstract. Xoring two permutations is a very simple way to construct pseudorandom functions from pseudorandom permutations. The aim of this paper is to get precise security results for this construction when the two permutations on n bits f and g are public. We will first prove that $f \oplus g$ is indifferentiable from a random function on n bits when the attacker is limited with q queries, with $q \ll \sqrt{2^n}$. This bound is called the “birthday bound”. We will then prove that this bound can be improved to $q^3 \ll 2^{2n}$. We essentially instantiate length preserving random functions, starting from fixed key ideal cipher with high security guarantee.

Key words: Indifferentiability, Luby-Rackoff Backwards with public permutations, Building random oracles from ideal block ciphers.

1 Introduction

The theme of this paper is to prove some security bounds about the indifferentiability of the Xor of two public random permutations on n bits from one random public function on n bits. We will look for security bounds “beyond the birthday bound” and smaller than the “information bound”, i.e. when the number of queries q is $q \ll 2^n$, but we may have $q \gg \sqrt{2^n}$. Therefore, this paper is in relation with previous work about the Xor of two permutations, about previous work dealing with security proofs “beyond the birthday bound” for various ideal cryptographic constructions, and, of course, about previous work on indifferentiability.

Luby-Rackoff Backwards.

The problem to construct pseudorandom functions (PRF) from pseudorandom permutations (PRP) is called “Luby-Rackoff Backwards”. This problem was first considered in [3]. This problem is obvious if the pseudorandom permutations are secret and if we are interested in an asymptotical polynomial versus non polynomial model (since a PRP is then a PRF). However, this problem is not obvious if

we want security beyond the birthday bound, or if the permutations are public. When the permutations are secret, Lucks ([11]) has proved that the Xor of k independent pseudorandom permutations gives security when $q \ll 2^{\frac{k}{k+1}n}$. (For $k = 2$ this gives $O(2^{\frac{2}{3}n})$). This bound was improved in [2, 19] and [20] where proofs of security for $q \ll 2^n$ are given. (However, Lucks proof is much simpler). When $q = 2^n$, as pointed out in these papers, it is easy to distinguish $\pi_1 \oplus \pi_2$ from a random functions R since $\oplus_{x \in \{0,1\}^n} (\pi_1 \oplus \pi_2)(x) = 0$. In this paper π_1 and π_2 will be public, and therefore our bounds are necessary smaller than the bounds obtained when π_1 and π_2 are secret. We will in fact match the original bound proven by Lucks, i.e. $q \ll 2^{\frac{2}{3}n}$.

Security proofs beyond the birthday bound.

Many papers have been published with security proofs beyond the birthday bound for various ideal cryptographic constructions. For example Aiello and Verkatesan [1] for doubling the length of pseudorandom functions with the Benes construction, or Maurer and Prietzkak [12] or Patarin [16, 17] for Feistel schemes.

Indifferentiability.

However the main topic of this paper is related to indifferentiability theory since, again, π_1 and π_2 will be public. The notion of indifferentiability was introduced by Maurer, Renner and Holenstein [13]. Since then, a lot of works has been done about indifferentiability. For example, in [6], Coron et al have shown how to construct a random oracle from an ideal block cipher. Their proved security bound is in $q \ll \sqrt{2^n}$, where n is the number of bits of the output of the ideal cipher. With our construction we will also be able to construct random oracles from ideal block ciphers. The other direction (constructing an ideal cipher from an oracle model) was proved in [7]. Their construction uses a 6-round Feistel scheme, and the security is proved for $q^{16} \ll 2^n$. This is below the birthday bound ($q \ll \sqrt{2^n}$). In his PhD, [22], Seurin has obtained a better bound, $q^4 \ll 2^n$, but for more rounds: 10 rounds instead of 6. This bound is better but still below the birthday bound. Our problem is different and simpler, this is why we will be able to obtain better security bounds.

Related Works

Recently Dodis *et al* [8], has shown XOR of a random permutation and its inverse is actually indifferentiable from a random function. However, they achieved a birthday security bound. Whereas, we show with XOR of two independent random permutations one can get a security proof with beyond birthday security guarantee.

2 Bounding Distinguisher’s Advantage

A *distinguisher* (attacker) D , for two oracles \mathcal{F} and \mathcal{G} , is an algorithm which has access to either oracle \mathcal{F} or oracle \mathcal{G} and outputs either 0 or 1 after making queries to the given oracle. The *advantage* $\text{Adv}_D(\mathcal{F}, \mathcal{G})$ or simply Adv_D of the distinguisher D is defined as

$$\text{Adv}_D = |\Pr[D^{\mathcal{F}} \rightarrow 1] - \Pr[D^{\mathcal{G}} \rightarrow 1]|.$$

The *view* V of the distinguisher is nothing but the list of the queries made by and the responses it received from the given oracle. \mathcal{V}^F and \mathcal{V}^G be the random variables corresponding to the distinguisher's view, when D is interacting with \mathcal{F} and \mathcal{G} respectively. \mathcal{V} be the set of all possible views. One can actually easily show [4, 5, 15],

$$\text{Adv}_D \leq \frac{1}{2} \sum_{V \in \mathcal{V}} |\Pr[\mathcal{V}^F = V] - \Pr[\mathcal{V}^G = V]|.$$

Below, we state two well known theorems [4, 5, 15] regarding upper bounds on Adv_D .

Theorem 1. *If for all $V \in \mathcal{V}$, we have*

$$\Pr[\mathcal{V}^G = V] \geq (1 - \varepsilon) \Pr[\mathcal{V}^F = V],$$

then $\text{Adv}_D \leq \varepsilon$.

Theorem 2. *If for all $V \in \mathcal{M} \subseteq \mathcal{V}$, we have*

$$\Pr[\mathcal{V}^G = V] \geq (1 - \varepsilon_1) \Pr[\mathcal{V}^F = V]$$

and $\Pr[\mathcal{V}^F \notin \mathcal{M}] \leq \varepsilon_2$, then $\text{Adv}_D \leq \varepsilon_1 + \varepsilon_2$.

3 Indifferentiability

The notion of indifferentiability was introduced by Maurer, Renner and Holenstein in [13]. This is an extension of the classical notion of indistinguishability, where one or more oracles are publicly available, such as random oracles, random permutations, or ideal ciphers. This notion of indifferentiability is used to show that an ideal primitive \mathcal{G} (for example a random function) can be replaced by a construction C that is based on some other ideal primitive \mathcal{F} (for example, C is the Xor of two random permutations).

Definition 1. Indifferentiability [13]

A Turing machine C with oracle access to an ideal primitive \mathcal{F} is said to be $(t, q_C, q_{\mathcal{F}}, \varepsilon)$ indifferentiable from an ideal primitive \mathcal{G} if there exists a simulator S with an oracle access to \mathcal{G} and running time at most t , such that for any distinguisher D , it holds that

$$\text{Adv}_D((C^{\mathcal{F}}, \mathcal{F}), (\mathcal{G}, S^{\mathcal{G}})) < \varepsilon.$$

The distinguisher makes at most q_C queries to C or \mathcal{G} and at most $q_{\mathcal{F}}$ queries to \mathcal{F} or S . Similarly, $C^{\mathcal{F}}$ is said to be (computationally) indifferentiable from \mathcal{G} if running time of D is bounded above by some polynomial in the security parameter k and ε is a negligible function of k .

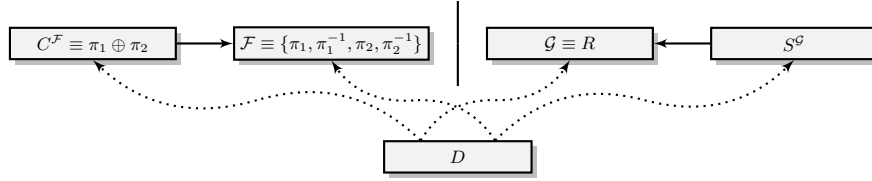


Fig. 1. The indistinguishability notion

The previous definition is illustrated in Figure 1. $I_n = \{0, 1\}^n$ denotes the set of all bit strings of length n . As in this paper, $R : I_n \rightarrow I_n$ is a random function, $\pi_1, \pi_2 : I_n \rightarrow I_n$ are two random permutations, and C is $\pi_1 \oplus \pi_2$. The distinguisher has either access to the system formed by the construction C and π_1, π_2 , or to the system formed by the random function R and a simulator S . In the first system (left), the construction C computes its output by making calls to π_1, π_2, π_1^{-1} and π_2^{-1} . In our case $C^{\{\pi_1, \pi_2\}}$ is just $\pi_1 \oplus \pi_2$, so in our case C does not need access to π_1^{-1}, π_2^{-1} but just π_1, π_2 . The distinguisher can also make calls to π_1, π_2, π_1^{-1} and π_2^{-1} directly. In the second system (right), the distinguisher can either query the random function R , or the simulator S that can make query to R . We see that the role of the simulator is to simulate the random permutations π_1, π_2 and to simulate also π_1^{-1}, π_2^{-1} , such that no distinguisher can tell whether it is interacting with C and π_1, π_2, π_1^{-1} and π_2^{-1} , or with R and S . Notice that the simulator does not see the distinguisher's queries to π . However, it can call R directly when needed for the simulation. The output of S should be indistinguishable from that of random oracle permutations π_1, π_2 , and the output of S should look consistent with what the distinguisher can obtain from R .

4 Our simulator

S denotes the simulator, and D the distinguisher. After α queries, S maintains always the sequence (x_i, a_i, b_i) , $1 \leq i \leq \alpha$, containing previous responses, as we will see, with $\forall i, 1 \leq i \leq \alpha, a_i \oplus b_i = R(x_i)$. When D contacts R , he makes only direct queries: D gives a value x'_i to R , and obtains the value $R(x'_i)$. S does not know these values x'_i . When D contacts S , we can assume without losing generality that D can make only 3 types of queries: A direct query, or an inverse query with a_α , or an inverse query with b_α .

Direct query.

In a direct query, D gives a new value x_α to S (i.e. $x_\alpha \notin \{x_1, \dots, x_{\alpha-1}\}$) and S will give to D a value a_α to simulate $\pi_1(x_\alpha)$ and a value b_α to simulate $\pi_2(x_\alpha)$. We can assume, without losing generality that D chooses $x_\alpha \notin \{x_1, \dots, x_{\alpha-1}\}$ because if $x_\alpha = x_i, i \leq \alpha - 1$, then S will always answer $a_\alpha = a_i$ and $b_\alpha = b_i$ and D will learn nothing new. Our simulator will compute a_α and b_α like this:

1. S asks for the value $R(x_\alpha)$.

2. a_α is randomly chosen with a uniform distribution in $I_n \setminus \{a_1, a_2, \dots, a_{\alpha-1}, R(x_\alpha) \oplus b_1, R(x_\alpha) \oplus b_2, \dots, R(x_\alpha) \oplus b_{\alpha-1}\}$.
3. $b_\alpha = R(x_\alpha) \oplus a_\alpha$

We will denote $V_{\alpha-1} = \{a_1, a_2, \dots, a_{\alpha-1}\}$ and $Q_{\alpha-1} = \{R(x_\alpha) \oplus b_1, R(x_\alpha) \oplus b_2, \dots, R(x_\alpha) \oplus b_{\alpha-1}\}$. Therefore we will have: $Q_{\alpha-1} = \{b_\alpha \oplus a_\alpha \oplus b_1, b_\alpha \oplus a_\alpha \oplus b_2, \dots, b_\alpha \oplus a_\alpha \oplus b_{\alpha-1}\}$, and a_α is randomly chosen in $I_n \setminus (V_{\alpha-1} \cup Q_{\alpha-1})$.

Inverse query with a_α .

In such inverse query, D gives a new value a_α to S (i.e. $a_\alpha \notin \{a_1, a_2, \dots, a_{\alpha-1}\}$) and S will give to D a value x_α to simulate $\pi_1^{-1}(a_\alpha)$ and a value b_α to simulate $\pi_2(x_\alpha)$. We can assume, without losing generality that D chooses $a_\alpha \notin \{a_1, a_2, \dots, a_{\alpha-1}\}$ because if $a_\alpha = a_i$, $i \leq \alpha - 1$, then S will always answer $x_\alpha = x_i$ and $b_\alpha = b_i$ and D will learn nothing new. Our simulator will compute x_α and b_α like this:

1. x_α is randomly chosen with a uniform distribution in $I_n \setminus \{x_1, \dots, x_{\alpha-1}\}$.
2. S asks for the value $R(x_\alpha)$.
3. If $R(x_\alpha) \oplus a_\alpha \notin \{b_1, b_2, \dots, b_{\alpha-1}\}$ then S gives this x_α to D and gives $b_\alpha = R(x_\alpha) \oplus a_\alpha$ to D .
4. If $R(x_\alpha) \oplus a_\alpha \in \{b_1, b_2, \dots, b_{\alpha-1}\}$ then S goes back to 1 above, and tries with another x_α randomly chosen in $I_n \setminus \{x_1, \dots, x_{\alpha-1}\}$.

This process continues until S has found like this a value x_α such that $R(x_\alpha) \oplus a_\alpha \notin \{b_1, b_2, \dots, b_{\alpha-1}\}$ and then it gives this x_α to D and $b_\alpha = R(x_\alpha) \oplus a_\alpha$. If S cannot find such a x_α it does not answer, but in general this probability will be negligible if $\alpha \ll 2^n$. Therefore, when S answers, the value x_α has been randomly chosen with a uniform distribution in $I_n \setminus \{x_1, \dots, x_{\alpha-1}\} \setminus W_{\alpha-1}$ where $W_{\alpha-1} = \{x \in I_n \text{ such that } R(x) \oplus a_\alpha \in \{b_1, b_2, \dots, b_{\alpha-1}\}\}$

Remark. A variant would be to choose a simulator S that will abort after k failed x_α , where k is an integer. In this paper we can actually assume $k = 2$, as we are looking for a security proof when $q \ll 2^{\frac{2n}{3}}$. We do not assume that D makes “timing attacks” but only that D computes from the values given by S without using the time for S to give them. However it would not change anything to assume that D tries to use this time since when S answers a value x_α , whatever the time S has used to compute x_α , x_α is always randomly chosen in $I_n \setminus \{x_1, \dots, x_{\alpha-1}\} \setminus W_{\alpha-1}$ and D knows this set $I_n \setminus \{x_1, \dots, x_{\alpha-1}\} \setminus W_{\alpha-1}$. Therefore the time gives no more information to D .

Inverse query with b_α .

In such inverse query, D gives a new value b_α to S (i.e. $b_\alpha \notin \{b_1, \dots, b_{\alpha-1}\}$) and S will give to D a value x_α to simulate $\pi_2^{-1}(b_\alpha)$ and a value a_α to simulate $\pi_1(x_\alpha)$. Our simulator will compute x_α in a symmetric way as we have just seen for inverse query with a_α . This means that S will randomly choose x_α with uniform distribution in $I_n \setminus \{x_1, \dots, x_{\alpha-1}\} \setminus W'_{\alpha-1}$ with $W'_{\alpha-1} = \{x \in I_n \text{ such that } R(x) \oplus b_\alpha \in \{a_1, a_2, \dots, a_{\alpha-1}\}\}$ and that S will give this x_α to D and S will give a_α to D with $a_\alpha = R(x_\alpha) \oplus b_\alpha$.

Whatever the query of D is, direct, inverse with a_α , or inverse with b_α , S will

store and memorize the values $(x_\alpha, a_\alpha, b_\alpha)$ generated to the sequence (x_i, a_i, b_i) , $1 \leq i \leq q$.

5 Distinguisher Characterization

Distinguisher D'

For any distinguisher D , with q queries, we consider another distinguisher D' with q' queries, $q' \geq q$ such that:

1. The first q queries of D' are exactly those of D .
2. D' outputs the same decision, 0 or 1, as D .
3. For any direct query x that D makes directly to R , D' will make **at the end** a direct query with this value x to S .

We can assume that D' does not make any duplicate query, since S and R will always give the same answers on the same questions. Since D and D' always output the same decision, we have: $\text{Adv}_D = \text{Adv}_{D'}$. If q is the number of queries that D makes to S or R , with q_1 the number of queries that D makes to R and q_2 the number of queries that D makes to S , then $q = q_1 + q_2$, $q' = q + q_1$, D' makes q_1 queries to R (as D) and $q = q_1 + q_2$ queries to S .

Let, $T = ((x_1, a_1, b_1), \dots, (x_q, a_q, b_q))$ be the view of the distinguisher D' . The i^{th} triple (x_i, a_i, b_i) implies D' received the triple during the i^{th} direct/inverse query to the Simulator or the Random permutations. B_n be the set of all permutations from I_n to I_n . D' tries to distinguish whether the sequence T , came from π_1 and π_2 with $\pi_1, \pi_2 \in_R B_n$ or from the simulator. p_T and p_T^* be the probabilities that D' receives the tuple T while interacting with the random permutations and simulator respectively. As D and D' output the same decision bit we have,

$$\text{Adv}_D = \text{Adv}_{D'} \leq \frac{1}{2} \sum_T |p_T - p_T^*|$$

Any sequence $T = \{(x_i, a_i, b_i), 1 \leq i \leq q\}$, such that $x_i, a_i, b_i \in I_n$, x_i 's are pairwise distinct, a_i 's are pairwise distinct and b_i 's are pairwise distinct as well is called *distinct* q -sequence. For any distinct q -sequence T we have,

$$p_T = \prod_{\alpha=1}^q \frac{1}{(2^n - (\alpha - 1))^2},$$

because π_1, π_2 are random permutations. For any other sequence T of q -triples p_T is zero.

6 Proof of security when $q \ll \sqrt{2^n}$

Theorem 3. π_1, π_2 be two random permutations and R be a random function $I_n \rightarrow I_n$. S be the simulator as defined before. Then for any distinguisher D , for the systems $(\pi_1 \oplus \pi_2, \{\pi_1, \pi_2\})$ and (R, S) making at most q queries we have, $\text{Adv}_D \leq \mathcal{O}\left(\frac{q^2}{2^n}\right)$

Proof. As discussed before, at first we construct the distinguisher D' starting from the distinguisher D . We know,

$$\text{Adv}_D = \text{Adv}_{D'}.$$

p_T and p_T^* are the probabilities that D' receives the distinct q -sequence $T = \{(x_i, a_i, b_i), 1 \leq i \leq q\}$ while interacting with the random permutations and simulator respectively. We already know,

$$p_T = \prod_{\alpha=1}^q \frac{1}{(2^n - (\alpha - 1))^2}.$$

Now, if we can show

$$p_T^* \geq (1 - \epsilon)p_T,$$

for all distinct q -sequence T , then by Theorem 1 that would imply

$$\text{Adv}_{D'} \leq \epsilon.$$

As we are interested in the lower bound of p_T^* , we evaluate p_T^* where we impose one extra condition, that is while answering the inverse queries the simulator never makes a bad guess. We define G_α to be the event, that this extra condition is satisfied during $(x_1, a_1, b_1), \dots, (x_\alpha, a_\alpha, b_\alpha)$ responses. Also, $(X_\alpha, A_\alpha, B_\alpha)$ be the random variable corresponding to α^{th} triple received by D' . We are interested in a lower bound for

$$P_\alpha \equiv \Pr[(X_\alpha, A_\alpha, B_\alpha) = (x_\alpha, a_\alpha, b_\alpha) \cap G_\alpha \mid (X_i, A_i, B_i) = (x_i, a_i, b_i) \text{ for } 1 \leq i \leq \alpha - 1 \cap G_{\alpha-1}].$$

Direct query with x_α :

We have $|V_{\alpha-1} \cup Q_{\alpha-1}| \geq (\alpha - 1)$. Hence,

$$P_\alpha \geq \frac{1}{2^n} \times \frac{1}{2^n - (\alpha - 1)}.$$

The term $\frac{1}{2^n}$, comes from the probability over the random function i.e. $R(x_\alpha) = a_\alpha \oplus b_\alpha$, the condition $G_{\alpha-1}$ implies $R(x_\alpha)$ was not queried beforehand.

Inverse query with a_α :

Conditioned on the event $G_{\alpha-1}$, the probability that the simulator guesses x_α in the first trial and $R(x_\alpha)$ outputs $a_\alpha \oplus b_\alpha$ is $\frac{1}{2^n - (\alpha - 1)} \times \frac{1}{2^n}$. For inverse query

with b_α we also get the same bound as above. So, whether α^{th} query is direct or inverse query we always have

$$P_\alpha \geq \frac{1 - \frac{\alpha-1}{2^n}}{(2^n - (\alpha - 1))^2}.$$

Hence,

$$p_T^* \geq p_T^{*'} = \prod_{\alpha=1}^q P_\alpha \geq \prod_{\alpha=1}^q \frac{1 - \frac{\alpha-1}{2^n}}{(2^n - (\alpha-1))^2} \geq p_T \left(1 - \frac{q^2}{2^n}\right).$$

This would imply $\text{Adv}_{D'} \leq \frac{q^2}{2^n}$. \square

7 Proof of security when $q \ll 2^{\frac{2}{3}n}$

Theorem 4. π_1, π_2 be two random permutations and R be a random function $I_n \rightarrow I_n$. S be the simulator as defined before. Then for any distinguisher D , for the systems $(\pi_1 \oplus \pi_2, \{\pi_1, \pi_2\})$ and (R, S) making at most q queries we have, $\text{Adv}_D \leq \mathcal{O}\left(\frac{q^3}{2^{2n}}\right)$

Proof. As discussed before, at first we construct the distinguisher D' starting from the distinguisher D . We know,

$$\text{Adv}_D = \text{Adv}_{D'}.$$

For any tuple $T = ((x_1, a_1, b_1), \dots, (x_q, a_q, b_q))$ for all the values $A \in I_n$, let $N_A(T)$ be the number of (i, j) , $1 \leq i \leq q$, $1 \leq j \leq q$ such that $a_i \oplus b_j = A$. M be the set of q -tuples such that $T \in M$ iff $N_A(T) \leq \frac{24q^2}{2^n - q}$ for all $A \in I_n$. p_T and p_T^* be the probabilities that D' receives the tuple $T = \{(x_i, a_i, b_i), 1 \leq i \leq q\}$ while interacting with the random permutations and simulator respectively. We already know,

$$p_T = \prod_{\alpha=1}^q \frac{1}{(2^n - (\alpha-1))^2}.$$

Theorem 5 from Appendix A implies,

$$\sum_{T \notin M} p_T \leq \frac{2^n}{2^{12n}} = \frac{1}{2^{11n}}.$$

Now, if we can show

$$p_T^* \geq (1 - \epsilon)p_T$$

for all $T \in M$, then by Theorem 2 that would imply

$$\text{Adv}_{D'} \leq \frac{1}{2^{11n}} + \epsilon.$$

While answering the inverse queries, it might be possible that the simulator makes some bad guess of x_α . As we are interested in the lower bound of p_T^* , we evaluate $p_T^{*'}$ where we impose some extra conditions, the simulator is allowed to make only one bad guess while answering inverse queries and the bad guess can not be same as any x_i for $1 \leq i \leq q$ or some previous bad guess. For $1 \leq \alpha \leq q$, we define G_α to be the event, that this extra condition is satisfied during $(x_1, a_1, b_1), \dots, (x_\alpha, a_\alpha, b_\alpha)$ responses. Also, $(X_\alpha, A_\alpha, B_\alpha)$ be the random

variable corresponding to α^{th} triple received by D' . We are interested in a lower bound for

$$P_\alpha \equiv \Pr[(X_\alpha, A_\alpha, B_\alpha) = (x_\alpha, a_\alpha, b_\alpha) \cap G_\alpha \\ |(X_i, A_i, B_i) = (x_i, a_i, b_i) \text{ for } 1 \leq i \leq \alpha - 1 \cap G_{\alpha-1}].$$

Direct query :

If $T \in M$, we have $|V_{\alpha-1} \cup Q_{\alpha-1}| \geq 2(\alpha - 1) - \frac{24q^2}{2^n - q}$. Hence, we have

$$P_\alpha \geq \frac{1}{2^n} \times \frac{1}{2^n - 2(\alpha - 1) + \frac{24q^2}{2^n - q}}.$$

The term $\frac{1}{2^n}$, comes from the probability over the random function i.e. $R(x_\alpha) = a_\alpha \oplus b_\alpha$. The condition $G_{\alpha-1}$, guarantees that x_α was not queried to the random oracle R , as a bad first guess in some previous inverse query. Assuming $q \leq 2^n/4$ we get,

$$P_\alpha \geq \frac{1 - \frac{96q^2}{2^{2n}}}{(2^n - (\alpha - 1))^2}.$$

Inverse query with a_α :

Conditioned on the event $G_{\alpha-1}$, the probability that the simulator guesses x_α in the first trial and $R(x_\alpha)$ outputs $a_\alpha \oplus b_\alpha$ is $\frac{1}{2^n - (\alpha - 1)} \times \frac{1}{2^n}$. Conditioned on the event $G_{\alpha-1}$, the probability that the simulator guesses x_α in the second trial satisfying $R(x_\alpha) = a_\alpha \oplus b_\alpha$ and condition G_α is at least,

$$\frac{2^n - q - (\alpha - 1)}{2^n - (\alpha - 1)} \times \frac{\alpha - 1}{2^n} \times \frac{1}{2^n - (\alpha - 1)} \times \frac{1}{2^n}.$$

$\frac{2^n - q - (\alpha - 1)}{2^n - (\alpha - 1)}$ corresponds to the probability that the first guess does not collide with x_1, \dots, x_q and the possible bad first guesses in the previous inverse queries. As the first guess is not queried before, $\frac{\alpha - 1}{2^n}$ is the probability that the first guess is bad. $\frac{1}{2^n - (\alpha - 1)} \times \frac{1}{2^n}$ is the probability that the second guess is x_α and $R(x_\alpha) = a_\alpha \oplus b_\alpha$. Hence all together we have,

$$P_\alpha \geq \frac{1}{2^n - (\alpha - 1)} \times \frac{1}{2^n} + \frac{2^n - q - (\alpha - 1)}{2^n - (\alpha - 1)} \times \frac{\alpha - 1}{2^n} \times \frac{1}{2^n - (\alpha - 1)} \times \frac{1}{2^n} \\ = \frac{1}{2^n} \times \frac{1}{2^n - (\alpha - 1)} \times \left(1 + \frac{2^n - q - (\alpha - 1)}{2^n - (\alpha - 1)} \times \frac{\alpha - 1}{2^n}\right)$$

Again assuming $q \leq 2^n/4$, we can show

$$P_\alpha \geq \frac{1 - \frac{4q^2}{2^{2n}}}{(2^n - (\alpha - 1))^2}.$$

For inverse queries with b_α , we also get the same lower bound as above. Hence, whether α^{th} query is direct or inverse query we always have

$$P_\alpha \geq \frac{1 - \frac{96q^2}{2^{2n}}}{(2^n - (\alpha - 1))^2}.$$

Hence,

$$p_T^* \geq p_T^{*'} = \prod_{\alpha=1}^q P_\alpha \geq \prod_{\alpha=1}^q \frac{1 - \frac{96q^2}{2^{2n}}}{(2^n - (\alpha - 1))^2} \geq p_T \left(1 - \frac{96q^3}{2^{2n}}\right).$$

This would imply $\text{Adv}_{D'} \leq \frac{1}{2^{11n}} + \frac{96q^3}{2^{2n}}$. □

8 Application of our Work

Even though the problem of constructing a public random function from public random permutations are interesting in its own right, here we briefly mention some possible application of our result. There are numerous cryptographic schemes [8, 14, 23, 24] where length preserving Random Functions are needed. Only known instantiation of those non-invertible length-preserving primitives were due to Dodis *et al* [8]. However, as stated below their instantiation does not always serve the purpose as the birthday security bound over there fails to preserve the high security of the constructions.

1. In Crypto 2007, Maurer and Tessaro [14] considered the problem of extending the domain of public random functions approaching optimal security bound, starting from length preserving random functions. With our result, if we choose to instantiate the random function as XOR of two fixed key ideal ciphers. Even though we won't be able to guarantee the optimal $\Theta(2^{n(1-\varepsilon)})$ security bound we can easily guarantee beyond birthday security bound up to $O(2^{\frac{2n}{3}})$ queries.
2. Stam in Crypto 2008 [24], Shrimpton and Stam in ICALP 2008 [23] considered the problem of building collision resistant compression functions starting from length preserving random functions. However, here whether we use our instantiation or instantiation due to Dodis *et al* [8] do not matter, because here the goal of their work is to achieve collision resistancy as close as the Birthday Barrier.

9 Conclusion

In this paper, we have proved the indistinguishability of the Xor of two random permutations on n bits from a random function on n bits when the number of

queries satisfies $q \ll \sqrt{2^n}$ (birthday bound) or $q \ll 2^{2n/3}$. The simulator S used was the same in both cases. In fact, it is conjectured that for this simulator the security is probably in $q \ll 2^n$, which if true would extend Maurer and Tessaro's [14] result preserving the optimal $\Theta(2^{n(1-\varepsilon)})$ bound.

Acknowledgements We sincerely thank Jean Sébastien Coron for his valuable comments and long discussions on initial drafts of this paper.

References

1. William Aiello and Ramarathnam Venkatesan. Foiling Birthday Attacks in Length-Doubling Transformations - Benes: A Non-Reversible Alternative to Feistel. In Ueli M. Maurer, editor, *Advances in Cryptology - EUROCRYPT 1996*, volume 1070 of *Lecture Notes in Computer Science*, pages 307–320. Springer-Verlag, 1996.
2. Mihir Bellare and Russell Impagliazzo. A Tool for Obtaining Tighter Security Analyses of Pseudorandom Function Based Constructions, with Applications to PRP to PRF Conversion. ePrint Archive 1999/024: Listing for 1999.
3. 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 1998*, volume 1403 of *Lecture Notes in Computer Science*, pages 266–280. Springer-Verlag, 1998.
4. Rishiraj Bhattacharyya, Avradip Mandal, and Mridul Nandi. Security analysis of the mode of jh hash function. In *FSE*, volume 6147 of *Lecture Notes in Computer Science*. Springer, 2010.
5. Donghoon Chang, Sangjin Lee, Mridul Nandi, and Moti Yung. Indifferentiable security analysis of popular hash functions with prefix-free padding. In *ASIACRYPT*, volume 4284 of *Lecture Notes in Computer Science*, pages 283–298. Springer, 2006.
6. J.C Coron, Y. Dodis, C. Malinaud, and P. Puniya. Merkle-Damgård Revisited: How to construct a Hash Function. In V. Shoup, editor, *Advances in Cryptology - CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 430–448. Springer-Verlag, 2005.
7. Jean-Sébastien Coron, Jacques Patarin, and Yannick Seurin. The Random Oracle Model and the Ideal Cipher Model are Equivalent. In David Wagner, editor, *Advances in Cryptology - CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 1–20. Springer-Verlag, 2008.
8. Yevgeniy Dodis, Krzysztof Pietrzak, and Prashant Puniya. A new mode of operation for block ciphers and length-preserving macs. In *Advances in cryptology - EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 198–219. Springer-Verlag, 2008.
9. Chris Hall, David Wagner, John Kelsey, and Bruce Schneier. Building PRFs from PRPs. In Hugo Krawczyk, editor, *Advances in Cryptology - CRYPTO 1998*, volume 1462 of *Lecture Notes in Computer Science*, pages 370–389. Springer-Verlag, 1998.
10. Marshall Hall Jr. A Combinatorial Problem on Abelian Groups. *Proceedings of the American Mathematical Society*, 3(4):584–587, 1952.
11. Stefan Lucks. The Sum of PRPs Is a Secure PRF. In Bart Preneel, editor, *Advances in Cryptology - EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 470–487. Springer-Verlag, 2000.

12. Ueli Maurer and Krzysztof Pietrzak. The Security of Many-Round Luby-Rackoff Pseudo-Random Permutations. In Eli Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 544–561. Springer-Verlag, 2003.
13. Ueli Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability Impossibility Results on Reductions, and Applications to the Random Oracle Methodology. In *TCC 2004*, volume 2951 of *Lecture Notes in Computer Science*, pages 21–39. Springer-Verlag, 2004.
14. Ueli M. Maurer and Stefano Tessaro. Domain extension of public random functions: Beyond the birthday barrier. In *CRYPTO*, volume 4622 of *Lecture Notes in Computer Science*, pages 187–204. Springer, 2007.
15. Mridul Nandi. A simple and unified method of proving indistinguishability. In *INDOCRYPT*, volume 4329 of *Lecture Notes in Computer Science*, pages 317–334. Springer, 2006.
16. Jacques Patarin. Luby-Rackoff: 7 Rounds are Enough for $2^{n(1-\epsilon)}$ Security. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 513–529. Springer-Verlag, 2003.
17. Jacques Patarin. On linear systems of equations with distinct variables and Small block size. In Dongho Wan and Seungjoo Kim, editors, *ICISC 2005*, volume 3935 of *Lecture Notes in Computer Science*, pages 299–321. Springer-Verlag, 2006.
18. Jacques Patarin. A proof of security in $O(2^n)$ for the Benes schemes. In Serge Vaudenay, editor, *Progress in Cryptology – AFRICACRYPT’ 08*, volume 5023 of *Lecture Notes in Computer Science*, pages 209–220. Springer-Verlag, 2008.
19. Jacques Patarin. A Proof of Security in $O(2^n)$ for the Xor of Two Random Permutations . In Reihaneh Safavi-Naini, editor, *ICITS 2008*, volume 5155 of *Lecture Notes in Computer Science*, pages 232–248. Springer-Verlag, 2008.
20. Jacques Patarin. A Proof of Security in $O(2^n)$ for the Xor of Two Random Permutations - Extended Version. *Cryptology ePrint archive: 2008/010: Listing for 2008*, 2008.
21. F. Salzborn and G. Szekeres. A Problem in Combinatorial Group Theory. *Ars Combinatoria*, 7:3–5, 1979.
22. Yannick Seurin. Primitives et Protocoles cryptographiques à sécurité prouvée. In *Ph. Thesis*. Université de Versailles - Saint Quentin – France, 2009.
23. Thomas Shrimpton and Martijn Stam. Building a collision-resistant compression function from non-compressing primitives. In *ICALP (2)*, volume 5126 of *Lecture Notes in Computer Science*, pages 643–654. Springer, 2008.
24. Martijn Stam. Beyond uniformity: Better security/efficiency tradeoffs for compression functions. In *CRYPTO*, volume 5157 of *Lecture Notes in Computer Science*, pages 397–412. Springer, 2008.

A Property of the $a_i \oplus b_j$ values

We will assume that $q^2 \geq n \cdot 2^n$.

Theorem 5. *For all A of I_n , if the values (a_1, a_2, \dots, a_q) are pairwise distinct and randomly chosen in I_n and if the values (b_1, b_2, \dots, b_q) are pairwise distinct and randomly chosen in I_n , then: the number N_A of (i, j) , $1 \leq i \leq q$, $1 \leq j \leq q$ such that $a_i \oplus b_j = A$ satisfies:*

$$\Pr[N_A \geq \frac{24q^2}{2^n - q}] \leq \frac{1}{2^{12n}}$$

This also means that the number U_A of $(a_1, a_2, \dots, a_q, b_1, b_2, \dots, b_q)$ such that the a_i are pairwise distinct, the b_i are pairwise distinct, and $N_A \geq \frac{24q^2}{2^n - q}$ satisfies: $U_A \leq \frac{1}{2^{12n}} [2^n(2^n - 1) \dots (2^n - q + 1)]^2$.

Remark. The coefficient $\frac{1}{2^{12n}}$ here is not very important, we can easily change it to another even smaller coefficient. What is important for us here is the $O(\frac{q^2}{2^n})$ value.

Proof of Theorem 5. When new values a_α and b_α are generated, the probability that $\exists j, j \leq \alpha - 1$ such that $a_\alpha \oplus b_j = A$ is $\leq \frac{\alpha - 1}{2^n - (\alpha - 1)}$ since we have $\alpha - 1$ values b_j , and since a_α is randomly generated in $I_n \setminus \{a_1, a_2, \dots, a_{\alpha-1}\}$. Similarly, the probability that $\exists j, j \leq \alpha - 1$ such that $a_j \oplus b_\alpha = A$ is $\leq \frac{\alpha - 1}{2^n - (\alpha - 1)}$, and the probability that $a_\alpha \oplus b_\alpha = A$ is $\leq \frac{1}{2^n - (\alpha - 1)}$. Therefore the probability that a_α and b_α will increase N_A from the values $a_i, b_i, 1 \leq i \leq \alpha - 1$ is $\leq \frac{2\alpha}{2^n - (\alpha - 1)}$. Moreover, if it occurs, then N_A will increase by at maximum 2 since if it exists j such that $a_\alpha \oplus b_j = A$ then j is unique because all the b_j values are pairwise distinct. If $N_A \geq 2k$, where k is an integer, then for at least k such values α , N_A was increased by at least 1, i.e. for at least k values α we had: $\exists j \leq \alpha - 1$ such that $a_\alpha \oplus b_j = A$, or $a_j \oplus b_\alpha = A$, or $a_\alpha \oplus b_\alpha = A$. Therefore,

$$\begin{aligned} \Pr[N_A \geq 2k] &\leq \binom{q}{k} \cdot \left(\frac{2q}{2^n - (q - 1)}\right)^k \\ \Pr[N_A \geq 2k] &\leq \frac{q!}{k!(q - k)!} \left(\frac{2q}{2^n - (q - 1)}\right)^k \\ \Pr[N_A \geq 2k] &\leq \frac{q^k}{k!} \left(\frac{2q}{2^n - q}\right)^k \\ \Pr[N_A \geq 2k] &\leq \frac{1}{k!} \left(\frac{2q^2}{2^n - q}\right)^k \quad (1) \end{aligned}$$

From Stirling formula, $k! \sim_{k \rightarrow +\infty} k^k e^{-k} \sqrt{2\pi k}$. If $k \geq \frac{12q^2}{2^n - q}$ this gives

$$k! \geq \left(\frac{12q^2}{e(2^n - q)}\right)^k \geq \left(\frac{4q^2}{2^n - q}\right)^k$$

Therefore, from (1), $\Pr[N_A \geq 2k] \leq \frac{1}{2^k}$ and since $k \geq \frac{12q^2}{2^n - q} \geq 12n$ we have: $\Pr[N_A \geq \frac{24q^2}{2^n - q}] \leq \frac{1}{2^{12n}}$ as claimed.