Counterexample to OWF Self-XOR Being a DOWF

Nathan Geier
Tel Aviv University
nathangeier@mail.tau.ac.il

Abstract

We study the effects of the XOR transformation, that is, $f^{\oplus 2}(x_1, x_2) \coloneqq f(x_1) \oplus f(x_2)$, on one-wayness. More specifically, we present an example showing that if one-way functions exist, there also exists a one-way function f such that $f^{\oplus 2}$ is not even a distributional one-way function, demonstrating that one-wayness may severely deteriorate.

1 Introduction

Arguably, the two most basic bit-string operations are concatenation and bit-wise XOR. Intuitively speaking, we understand that concatenation of several independent instances weakens "pseudo-randomness" (or indistinguishability when comparing two distributions) but improves one-wayness. We would like to think of the XOR as an opposing operation, one which improves "randomness" (or indistinguishability) but weakens one-wayness, though this is not exactly accurate. In contrast to the case of concatenation, one XOR is already enough to drastically improve "randomness" or destroy the OWF property altogether (examples will be given in Subsection 1.1). Here, we try to consider a more refined question: Perhaps applying the XOR transformation to a OWF does not necessarily result a OWF (which is folklore knowledge), but maybe some more delicate form of one-wayness remains. Distributional one-way functions, introduced by Impagliazzo and Luby [IL89], express a weaker notion of one-wayness where the adversary does not only need to supply some preimage, but rather, it needs to supply a uniform preimage (up to some small error). The intuition why this notion may be interesting here is that oftentimes, in such counterexamples, we invert the one-way function using somewhat "artificial" inputs, while the DOWF requirement forces us to output a preimage which is very close to uniform. It should be noted here that the existence of OWFs is equivalent to that of DOWFs, but nevertheless, not every DOWF is necessarily a OWF while the opposite does hold, so for specific functions this notion is indeed weaker. As it turns out, the folklore counterexample showing that applying XOR to a OWF does not necessarily produce a OWF, which one may stumble upon in an introductory cryptography course, is not easily generalized to the case where the resulted function is only claimed to be a DOWF. This appears to be a non-trivial twist on a very basic question. In this paper, we provide a more involved counterexample, showing that the XOR transformation applied to a OWF does not even have to produce a DOWF. Some of the techniques presented in this paper may be of independent use.

1.1 Basic Observations

Let us informally make a few basic observations, folklore knowledge or easy to see, supporting what we mentioned above regarding the XOR transformation. In the following, we let $f^{\oplus 2}(x_1, x_2) \coloneqq f(x_1) \oplus f(x_2)$.

Claim 1.1 (Being a OWF is not preserved)**.** *If OWFs exist, there exists a OWF f such that $f^{\oplus 2}(x_1, x_2)$ is not a OWF.*

Proof sketch. Let g be a OWF, w.l.o.g. length-doubling, and define

$$f(x, y, b) \coloneqq \begin{cases} y, g(x) & b = 0 \\ g(x), y & b = 1 \end{cases}$$

where $|y| = |g(x)|$. It is not hard to see that f is a OWF since a random y usually does not have a preimage, but we can easily invert every $z = z_1, z_2$ where $|z_1| = |z_2|$, w.r.t. $f^{\oplus 2}$, by choosing $b_1 = 0$, $b_2 = 1$, then choosing x_1, x_2 arbitrarily, and finally set $y_1 = z_1 \oplus g(x_2)$ and $y_2 = z_2 \oplus g(x_1)$. □

Remark 1.1. The XOR transformation may sometimes preserve being a OWF: If OWFs exist, there exists a OWF f such that $f^{\oplus 2}$ is a OWF. For example, choose a length-tripling PRG.

Notice that the counterexample no longer works if we only require $f^{\oplus 2}$ to be a distributional one-way function, since the inverter always outputs preimages with $b_1 \neq b_2$, and this only happens with probability half over a random input, so the statistical distance between $U, f(U)$ and $A(f(U)), f(U)$ must be at least half for our inverter A. In this paper we provide another counterexample to show that $f^{\oplus 2}$ is not necessarily a DOWF.

In the introduction, we also mentioned that one XOR may already be enough to drastically improve "randomness". Let us briefly elaborate, though this has no relevance to the rest of the paper:

Claim 1.2 ("Randomness" may get significantly amplified)**.** *There exists a random variable X over $\{0,1\}^{\mathrm{poly}(n)}$ that is $1 - \mathrm{neg}(n)$ far from uniform, such that $X_1 \oplus X_2$ is $\mathrm{neg}(n)$-close to uniform where X_1, X_2 are two independent draws from X.*

Proof sketch. Let X' be an r.v. resulted by drawing $x, y \sim U_{2n}$ and outputting $(x, y, \langle x, y \rangle)$. We have that X' is 1/2-far from uniform, but $X'_1 \oplus X'_2$ is $\mathrm{neg}(n)$-close to uniform. We choose X to be the concatenation of n independent instances of X'. □

Remark 1.2. The XOR transformation cannot create perfect uniformity out of thin air: For a random variable X, if $X_1 \oplus X_2$ is uniformly distributed then so must be X.

1.2 Results

Theorem 1.1 (Informal)**.** *If one-way functions exist, there also exists a one-way function f such that $f^{\oplus 2}(x_1, x_2)$ is not distributionally one-way.*

1.3 Overview

A naive approach of generalizing the counterexample would be, instead of choosing a bit b to decide the position of $g(x)$, to draw $i \leftarrow [p(n)]$ and output $y_1, \ldots, y_{i-1}, g(x), y_{i+1}, \ldots, y_{p(n)}$. This gives for every polynomial $p(n)$, a construction of a OWF f with a distributional inverter for $f^{\oplus 2}$ of statistical distance $1/p(n)$. However, the main issue remains: we only know how to output preimages where $i_1 \neq i_2$ but we have that $i_1 = i_2$ happens with inverse-polynomial probability. What we want is a single construction for f such that for every $q(n)$, we can distributionally invert $f^{\oplus 2}$ with statistical distance better than $1/q(n)$.

We are going to construct a OWF f such that $f^{\oplus 2}$ can be distributionally inverted completely (distance zero) in expected polynomial time $p(n)$, thus in particular, for every $q(n)$ we can terminate the inverter after $p(n) \cdot q(n)$ steps and have statistical distance at most $1/q(n)$ to its output. The main idea behind the construction of f is as follows: we are given w.l.o.g. a length-preserving OWF g. On input (x, y, π) where $|x| = n$, $|y| = n^2$ and π is interpreted as a random injection from $[n] \to [n^2]$, we set f's output to (y', π) where y' is computed from y by replacing for every i the $\pi(i)$'th bit of y by the i'th bit of $g(x)$. It is not hard to see why f is a OWF. Let us briefly explain the intuition why $f^{\oplus 2}$ can be distributionally inverted in expected polynomial time.

For an image $z = (y'_1 \oplus y'_2, \pi_1 \oplus \pi_2)$ of $f^{\oplus 2}$, imagine we are given $x^*_1, x^*_2, \pi^*_1, \pi^*_2$ drawn from the conditional distribution. Then we can easily output a continuation y^*_1, y^*_2 from the conditional distribution, which is just a random consistent continuation, by first choosing y^*_1 at random except for positions in $\mathrm{Im}(\pi^*_2) \setminus \mathrm{Im}(\pi^*_1)$ where it must be fixed according to the values we were given, to a XOR between appropriate positions of z and x^*_2. Next, y^*_2 is fixed in the only consistent way possible in positions $[n^2] \setminus \mathrm{Im}(\pi^*_2)$, according to a XOR between appropriate positions of z and x^*_1, y^*_1, while in $\mathrm{Im}(\pi^*_2)$ it is drawn at random. To conclude, if we are able to generate $x^*_1, x^*_2, \pi^*_1, \pi^*_2$ according to the conditional distribution, then we can also appropriately generate a continuation y^*_1, y^*_2.

For the next step, imagine we are only given π^*_1, π^*_2 drawn from the conditional distribution. Here we will be using standard rejection sampling: simply keep drawing pairs x^*_1, x^*_2 until we find a pair that is consistent with z, namely, the XOR between x^*_1 and x^*_2 at positions corresponding to the intersection $\mathrm{Im}(\pi^*_1) \cap \mathrm{Im}(\pi^*_2)$ is consistent with z. The idea here is that in expectation we draw up to $2^{|\mathrm{Im}(\pi^*_1) \cap \mathrm{Im}(\pi^*_2)|}$ times until we find a pair consistent in the intersection, and over random injections π^*_1, π^*_2 from $[n]$ to $[n^2]$, the expected value of this term is polynomially bounded. Note an important subtlety here: we generated a uniform consistent pair x^*_1, x^*_2, while a priory the conditional distribution may weigh them differently. However, since for every consistent pair x^*_1, x^*_2 the number of consistent continuations y^*_1, y^*_2 is exactly the same as it depends only on the cardinality of the intersection $\mathrm{Im}(\pi^*_1) \cap \mathrm{Im}(\pi^*_2)$, the conditional distribution of x^*_1, x^*_2 is simply uniform over consistent pairs.

Lastly, we need to generate π^*_1, π^*_2. Here we run into a problem: we cannot simply choose π^*_1 at random and set $\pi^*_2 = \pi^*_1 \oplus (\pi_1 \oplus \pi_2)$, because every consistent pair (π^*_1, π^*_2) may have a different number of continuations, so the conditional distribution is not necessarily uniform. (Though this may be resolved if g is assumed to be a one-way permutation.) Instead, we will add information to our OWF f so that π_1, π_2 may be reconstructed. As it turns out, there exists an efficient transformation E such that (a, b) can be efficiently reconstructed from $(a \oplus b, E(a) \oplus E(b))$ for $a \neq b$. We will use it by modifying the output to $(y', \pi, E(\pi))$.

2 Definitions

First, some basic definitions and notation: For a bit-string x, we denote by $x[i]$ the i'th bit of x, and similarly by $x[i : j]$ the sub-string between position i to j, ends included. For $n \in \mathbb{N}$, let $[n] \coloneqq \{1, \ldots, n\}$. We denote by U_n the uniform distribution over $\{0,1\}^n$. For distributions P, Q over Ω, we denote by $|P - Q|$ the total variation distance between them, that is, $\frac{1}{2} \sum_{\omega \in \Omega} |P(\omega) - Q(\omega)|$. In addition, we rely on the following standard computational concepts and notation:

- For $p : \mathbb{N} \to \mathbb{R}$, by $p = \mathrm{poly}(n)$, we mean that $p(n)$ is a polynomial in n.
- A function $\mu : \mathbb{N} \to [0, 1]$ is negligible, denoted $\mu = \mathrm{neg}(n)$, if for every $p = \mathrm{poly}(n)$ we have that $\mu(n) < 1/p(n)$ for large enough n's.
- A function $f : \{0,1\}^* \to \{0,1\}^*$ is poly-time computable if there exists $p = \mathrm{poly}(n)$ and a Turing machine T such that on every input x, T outputs $f(x)$ after at most $p(|x|)$ steps.

- A non-uniform PPT $A = \{A_n\}$ is a sequence of probabilistic circuits such that $\mathsf{size}(A_n) \leq p(n)$ for some $p = \mathrm{poly}(n)$.

Definition 2.1 (One-Way Function)**.** A poly-time computable $f : \{0,1\}^* \to \{0,1\}^*$ is a one-way function if for every n.u. PPT $A = \{A_n\}$ it holds that

$$\Pr_{y \leftarrow f(U_n)} \left[A_n(y) \in f^{-1}(y) \right] = \mathrm{neg}(n).$$

Definition 2.2 (Distributional One-Way Function)**.** A poly-time computable $f : \{0,1\}^* \to \{0,1\}^*$ is a distributional one-way function if there exists $p = \mathrm{poly}(n)$ such that for every n.u. PPT $A = \{A_n\}$, for large enough n's, it holds that

$$|(U_n, f(U_n) - A_n(f(U_n)), f(U_n))| > 1/p(n).$$

Definition 2.3 (The XOR Transformation)**.** Given $f : \{0,1\}^* \to \{0,1\}^*$, we define

$$f^{\oplus 2}(x) \coloneqq \begin{cases} f\left(x[1:n/2]\right) \oplus f\left(x[n/2+1:n]\right) & n = |x| \ \ \mathsf{is\ even} \\ f^{\oplus 2}(x[1:n-1]) & n = |x| \ \ \mathsf{is\ odd} \end{cases}$$

Remark 2.1. This definition naturally extends to $f^{\oplus k}$, which we do not study in this paper.

3 The Counterexample

Let us present the construction and proof more formally.

Theorem 3.1 (Main)**.** *If one-way functions exist, there also exists a one-way function f such that $f^{\oplus 2}$ is not a distributional one-way function.*

The Construction. Let g be a OWF, w.l.o.g. length-preserving. We define f on input (x, y, π) where $|x| = n$, $|y| = n^2$, $|\pi| = \mathrm{poly}(n)$ by interpreting π as an injection from $[n]$ to $[n^2]$ and setting the output to $(y', \pi, E(\pi))$ where $|y'| = |y|$ with

$$y'[i] = \begin{cases} g(x)[\pi^{-1}(i)] & i \in \mathrm{Im}(\pi) \\ y[i] & \mathsf{otherwise} \end{cases}$$

and E is an efficient transformation family such that for $a \neq b$ with $|a| = |b| = m$, we can efficiently reconstruct (a, b) from $(a \oplus b, E_m(a) \oplus E_m(b))$. More details on how E may be implemented are given in Lemma 3.3. Also note that defining f over strings of length $\ell(n)$ where $\ell = \mathrm{poly}(n)$ is done w.l.o.g. since we can always extend f by ignoring some (but up to $(1 - 1/\mathrm{poly})$ fraction) of the input bits and later on have the distributional inverter of $f^{\oplus 2}$ draw them uniformly.

Claim 3.1. *If g is a OWF, then f proposed above is also a OWF.*

Proof. An inverter for f immediately yields an inverter for g: given $g(x)$, draw y and π and compute $f(x, y, \pi)$ which only depends on x through g, and feed to the inverter of f. For any preimage (x', y', π') of $f(x, y, \pi)$ it must be that $\pi = \pi'$ and furthermore, for every $i \in \mathrm{Im}\,(\pi)$ we have $g(x)[\pi^{-1}(i)] = y'[i] = g(x')[\pi^{-1}(i)]$ which implies that $g(x) = g(x')$. □

The XOR. The function we are trying to invert, up to a poly number of ignored bits which we can later draw at random, is

$$f^{\oplus 2}(x_1, y_1, \pi_1, x_2, y_2, \pi_2) = f(x_1, y_1, \pi_1) \oplus f(x_2, y_2, \pi_2) = (y'_1 \oplus y'_2, \pi_1 \oplus \pi_2, E(\pi_1) \oplus E(\pi_2)).$$

To be more explicit, we note that

$$(y'_1 \oplus y'_2)[i] = \begin{cases} g(x_1)[\pi_1^{-1}(i)] \oplus g(x_2)[\pi_2^{-1}(i)] & i \in \mathrm{Im}(\pi_1) \cap \mathrm{Im}(\pi_2) \\ g(x_1)[\pi_1^{-1}(i)] \oplus y_2[i] & i \in \mathrm{Im}(\pi_1) \setminus \mathrm{Im}(\pi_2) \\ y_1[i] \oplus g(x_2)[\pi_2^{-1}(i)] & i \in \mathrm{Im}(\pi_2) \setminus \mathrm{Im}(\pi_1) \\ y_1[i] \oplus y_2[i] & \mathsf{otherwise} \end{cases}$$

The Adversary. First, A uses $(\pi_1 \oplus \pi_2, E(\pi_1) \oplus E(\pi_2))$ to reconstruct (π_1, π_2), except for the special case of $\pi_1 = \pi_2$ for which A simply keeps guessing $(\pi, x_1, x_2, y_1, y_2)$ until a preimage is found. After the reconstruction, A keeps drawing pairs x_1^*, x_2^* until a consistent pair is found, that is

$$\forall i \in \mathrm{Im}(\pi_1) \cap \mathrm{Im}(\pi_2) : \; g(x_1^*)[\pi_1^{-1}(i)] \oplus g(x_2^*)[\pi_2^{-1}(i)] = (y'_1 \oplus y'_2)[i].$$

Finally, A chooses a random consistent continuation y_1^*, y_2^* as follows:

$$(y_1^*[i], y_2^*[i]) = \begin{cases} (\mathsf{random\ bit}, \mathsf{random\ bit}) & i \in \mathrm{Im}(\pi_1) \cap \mathrm{Im}(\pi_2) \\ \big(\mathsf{random\ bit}, \; g(x_1^*)[\pi_1^{-1}(i)] \oplus (y'_1 \oplus y'_2)[i]\big) & i \in \mathrm{Im}(\pi_1) \setminus \mathrm{Im}(\pi_2) \\ \big((y'_1 \oplus y'_2)[i] \oplus g(x_2^*)[\pi_2^{-1}(i)], \; \mathsf{random\ bit}\big) & i \in \mathrm{Im}(\pi_2) \setminus \mathrm{Im}(\pi_1) \\ (\mathsf{random\ bit}, \; \mathsf{same\ random\ bit} \oplus (y'_1 \oplus y'_2)[i]) & \mathsf{otherwise} \end{cases}$$

We want to show that A produces the correct distribution, and also that its running time is polynomially bounded in expectation over a random input to $f^{\oplus 2}$ and A's random coins.

Claim 3.2. *The output of A is a random preimage of $f^{\oplus 2}$.*

Proof. This is straightforward in the case of $\pi_1 = \pi_2$. In the general case, this can be shown by noting these two points:

- A always terminates when given a legal image, and whenever it does, it outputs a legal preimage. This follows readily from a direct inspection.
- Given any image $z = (z_1, z_2, z_3)$ as input, the probability for A to output any legal preimage $(x_1, y_1, \pi_1, x_2, y_2, \pi_2)$ is the same and depends only on z. We have that π_1, π_2 are determined by z_2, z_3 and chosen by A with probability 1. It must be that x_1, x_2 satisfy the consistent pair condition so A then chooses them with probability

$$\frac{1}{\left|\left\{x_1, x_2 \,|\, \forall i \in \mathrm{Im}(\pi_1) \cap \mathrm{Im}(\pi_2) : \; g(x_1)[\pi_1^{-1}(i)] \oplus g(x_2)[\pi_2^{-1}(i)] = z_1[i]\right\}\right|}$$

which only depends on z_1, π_1, π_2 and thus only on z_1, z_2, z_3. Finally, from the definition of $f^{\oplus 2}$, the following conditions must be satisfied by y_1, y_2:

$$\begin{aligned}
&\forall i \in \mathrm{Im}(\pi_2) \setminus \mathrm{Im}(\pi_1) :\ y_1[i] = g(x_2)[\pi_2^{-1}(i)] \oplus z_1[i] \\
&\forall i \in \mathrm{Im}(\pi_1) \setminus \mathrm{Im}(\pi_2) :\ y_2[i] = g(x_1)[\pi_1^{-1}(i)] \oplus z_1[i] \\
&\forall i \notin \mathrm{Im}(\pi_1) \cup \mathrm{Im}(\pi_2) :\ y_2[i] = y_1[i] \oplus z[i]
\end{aligned}$$

Hence A outputs y_1, y_2 if it guesses correctly the values of $\{y_1[i]\}_{i \in \overline{\mathrm{Im}(\pi_2) \setminus \mathrm{Im}(\pi_1)}}$ and $\{y_2[i]\}_{i \in \mathrm{Im}(\pi_2)}$ which happens with probability

$$\left(\frac{1}{2}\right)^{\left|\overline{\mathrm{Im}(\pi_2) \setminus \mathrm{Im}(\pi_1)}\right| + |\mathrm{Im}(\pi_2)|} = \left(\frac{1}{2}\right)^{n^2 - n + |\mathrm{Im}(\pi_1) \cap \mathrm{Im}(\pi_2)| + n} = \left(\frac{1}{2}\right)^{n^2 + |\mathrm{Im}(\pi_1) \cap \mathrm{Im}(\pi_2)|}$$

that depends only on π_1, π_2 and thus only on z_2, z_3.

Therefore, A outputs a uniform preimage. □

Claim 3.3. *The expected running time of A, over a random input to $f^{\oplus 2}$ and the randomness of A, is polynomially bounded.*

Proof. We handle the special case of $\pi_1 = \pi_2$ at the end. For the general case, reconstructing π_1, π_2 at the start and drawing y_1^*, y_2^* at the end are done efficiently, so the term we are left to bound is the number of draws it takes to succeed in finding a consistent x_1^*, x_2^* pair, that is, a pair where

$$\forall i \in \mathrm{Im}(\pi_1) \cap \mathrm{Im}(\pi_2) :\ g(x_1^*)[\pi_1^{-1}(i)] \oplus g(x_2^*)[\pi_2^{-1}(i)] = g(x_1)[\pi_1^{-1}(i)] \oplus g(x_2)[\pi_2^{-1}(i)].$$

For any fixed choice of (π_1, π_2), let $i_1 < \cdots < i_k$ be the elements of $\mathrm{Im}(\pi_1) \cap \mathrm{Im}(\pi_2)$ in an increasing order, and define $\mathrm{Proj}_{(\pi_1,\pi_2)} : \{0,1\}^{2n} \to \{0,1\}^{|\mathrm{Im}(\pi_1) \cap \mathrm{Im}(\pi_2)|}$ using

$$\mathrm{Proj}_{(\pi_1,\pi_2)}(x_1, x_2) = \left(g(x_1)[\pi_1^{-1}(i_1)] \oplus g(x_2)[\pi_2^{-1}(i_1)]\right), \ldots, \left(g(x_1)[\pi_1^{-1}(i_k)] \oplus g(x_2)[\pi_2^{-1}(i_k)]\right)$$

Then equivalently, the consistency condition can be formulated as $\mathrm{Proj}_{(\pi_1,\pi_2)}(x_1^*, x_2^*) = \mathrm{Proj}_{(\pi_1,\pi_2)}(x_1, x_2)$. By considering $\mathrm{Proj}_{(\pi_1,\pi_2)}(U_{2n})$ as our random variable, we are exactly in the setting of Lemma 3.2 given below, and we can use it to conclude that for every choice of (π_1, π_2) the expected number of draws, over both X_1, X_2 and the randomness of A, is

$$\left|\mathrm{Supp}\left(\mathrm{Proj}_{(\pi_1,\pi_2)}(U_{2n})\right)\right| = \left|\mathrm{Im}\left(\mathrm{Proj}_{(\pi_1,\pi_2)}\right)\right| \leq 2^{|\mathrm{Im}(\pi_1) \cap \mathrm{Im}(\pi_2)|}.$$

Finally, we bound $\mathbb{E}_{\pi_1,\pi_2}\left[2^{|\mathrm{Im}(\pi_1) \cap \mathrm{Im}(\pi_2)|}\right] \leq e^3$ using Lemma 3.1 which is given below. Note that when interpreting a long enough random string as an injection from $[n]$ to $[n^2]$, we don't get exactly a uniform injection but we can get something very close, say, completely uniform unless some failure happened with probability $2^{-\delta \cdot n^2}$. But $2^{|\mathrm{Im}(\pi_1) \cap \mathrm{Im}(\pi_2)|}$ is always upper bounded by 2^n, so the total expectation is at most $1 \cdot e^3 + \left(2 \cdot 2^{-\delta \cdot n^2}\right) \cdot 2^n \leq e^4$ for large enough n's. The case of $\pi_1 = \pi_2$ (representation-wise, not just functionality-wise) cannot significantly increase the expected running time because it happens with probability $2^{-|\pi|}$, and the expected running time in this case, using Lemma 3.2, is at most $2^{|x|+|y|}$. We can always add dummy bits to π in order to make sure that $|\pi| > |x| + |y|$. □

Proof of Theorem 3.1. Given a OWF g, we construct f as described above. The theorem then follows from claims 3.1, 3.2 and 3.3, coupled with the fact that we can stop the execution of A after $q(n)p(n)$ steps where $p(n)$ is the expected running time of A, and have statistical distance of at most $1/q(n)$ to A's output, which is a uniformly distributed preimage. □

Lemma 3.1. *Let π_1, π_2 be uniformly and independently chosen injections from $[n]$ to $[n^2]$, then we have that $\mathbb{E}_{\pi_1,\pi_2}\left[2^{|\mathrm{Im}(\pi_1)\cap\mathrm{Im}(\pi_2)|}\right] \leq e^3$.*

Remark 3.1. The bound also holds when we do not insist that π_1, π_2 are injections, because collisions can only decrease the image.

Proof. First, note that

$$\mathbb{E}_{\pi_1,\pi_2}\left[2^{|\mathrm{Im}(\pi_1)\cap\mathrm{Im}(\pi_2)|}\right] = \sum_{k=0}^{n} 2^k \Pr_{\pi_1,\pi_2}\left[|\mathrm{Im}(\pi_1)\cap\mathrm{Im}(\pi_2)| = k\right] = \sum_{k=0}^{n} 2^k \frac{\binom{n}{k}\binom{n^2-n}{n-k}}{\binom{n^2}{n}}$$

where we used that

$$\Pr_{\pi_1,\pi_2}\left[|\mathrm{Im}(\pi_1)\cap\mathrm{Im}(\pi_2)| = k\right] = \frac{\binom{n}{k}\binom{n^2-n}{n-k}n!}{\binom{n^2}{n}n!}$$

because for any π_1 the number of π_2's with intersection k is given by choosing k elements from $\mathrm{Im}(\pi_1)$ and another $n-k$ from $\overline{\mathrm{Im}(\pi_1)}$, then ordering them. Next, using $n^k \leq \frac{(n+k)!}{n!} \leq (n+k)^k$, we bound

$$\frac{\binom{n^2-n}{n-k}}{\binom{n^2}{n}} = \frac{(n^2-n)!\,(n^2-n)!n!}{(n-k)!\,(n^2-2n+k)!\,(n^2)!} = \frac{(n^2-n)!}{(n^2)!}\cdot\frac{(n^2-n)!}{(n^2-2n+k)!}\cdot\frac{n!}{(n-k)!} \leq$$

$$\leq \left(n^2-n\right)^{-n}\cdot\left(n^2-n\right)^{n-k}\cdot n^k = \left(\frac{n}{n^2-n}\right)^k = \left(\frac{1}{n-1}\right)^k$$

Plugging it back, we get

$$\mathbb{E}_{\pi_1,\pi_2}\left[2^{|\mathrm{Im}(\pi_1)\cap\mathrm{Im}(\pi_2)|}\right] \leq \sum_{k=0}^{n} 2^k \binom{n}{k}\left(\frac{1}{n-1}\right)^k = \sum_{k=0}^{n}\binom{n}{k}\left(\frac{2}{n-1}\right)^k 1^{n-k} =$$

$$= \left(1+\frac{2}{n-1}\right)^n \leq e^{2n/(n-1)} \leq e^3$$

Where we used the Binomial theorem and $1+x \leq e^x$. We also used that $2n/(n-1) \leq 3$ for $n \geq 3$. For $n < 3$, we have $2^{|\mathrm{Im}(\pi_1)\cap\mathrm{Im}(\pi_2)|} \leq 2^n \leq e^n \leq e^3$. □

Lemma 3.2. *Let X be any random variable. Consider drawing samples $(x, x_1, \ldots, x_i, \ldots)$ independently from X, and define the random variable N to be the first i such that $x_i = x$. Then, $\mathbb{E}[N] = |\mathrm{Supp}(X)|$.*

Proof. We denote by N_x the random variable N conditioned on the first draw being x. Notice that N_x is geometrically distributed with parameter $p_x := \Pr[X = x]$, and thus $\mathbb{E}[N_x] = 1/p_x$. We have that

$$\mathbb{E}[N] = \mathbb{E}_{x\leftarrow X}\left[\mathbb{E}[N_x]\right] = \mathbb{E}_{x\leftarrow X}\left[\frac{1}{p_x}\right] = \sum_{x\in\mathrm{Supp}(X)} p_x\cdot\frac{1}{p_x} = |\mathrm{Supp}(X)|.$$

□

Lemma 3.3 (See [Lin69])**.** *There exists an efficient transformation $E : \{0,1\}^n \to \{0,1\}^n$ such that for any $x, y \in \{0,1\}^n$ with $x \neq y$, we can efficiently reconstruct (x, y) from $(x \oplus y, E(x) \oplus E(y))$.*

Proof. Given input $x \in \{0,1\}^n$, E interprets it as a field element over $\mathrm{GF}(2^n)$ and maps it to x^3. Given $(u, v) = (x + y, x^3 + y^3)$ for $x \neq y$, the reconstruction works as follows: We start by computing $u^2 + v/u = xy = x\,(x + u)$, where division by u is legal since $x \neq y$, and the first equality is true since

$$u^3 = x^3 + x^2 y + xy^2 + y^3 = \left(x^3 + y^3\right) + xy\,(x + y) = v + xy \cdot u.$$

We conclude that $x^2 + ux + (u^2 + u/v) = 0$. Notice that any quadratic polynomial over $\mathrm{GF}(2^n)$ is actually a linear transformation since $(a + b)^2 = a^2 + b^2$, so solving it translates to solving a set of linear equations. Further, note that any quadratic polynomial over a field has at most two roots, and in our case the roots are exactly x and y since they are different and the same quadratic equation holds for y from symmetry. The reconstruction is efficient since basic arithmetic over $\mathrm{GF}(2^n)$ is efficiently computable, and solving our set of linear equations can also be done efficiently. □

References

[IL89] R. Impagliazzo and M. Luby. One-way functions are essential for complexity based cryptography. In *30th Annual Symposium on Foundations of Computer Science*, pages 230–235, 1989.

[Lin69] Bernt Lindström. Determination of two vectors from the sum. *Journal of Combinatorial Theory*, 6(4):402–407, 1969.