On the Streaming Indistinguishability of a Random Permutation and a Random Function

Itai Dinur

Department of Computer Science, Ben-Gurion University, Israel

Abstract. An adversary with S bits of memory obtains a stream of Q elements that are uniformly drawn from the set $\{1, 2, \ldots, N\}$, either with or without replacement. This corresponds to sampling Q elements using either a random function or a random permutation. The adversary's goal is to distinguish between these two cases.

This problem was first considered by Jaeger and Tessaro (EUROCRYPT 2019), which proved that the adversary's advantage is upper bounded by $\sqrt{Q \cdot S/N}$. Jaeger and Tessaro used this bound as a streaming switching lemma which allowed proving that known time-memory tradeoff attacks on several modes of operation (such as counter-mode) are optimal up to a factor of $O(\log N)$ if $Q \cdot S \approx N$. However, if $Q \cdot S \ll N$ there is a gap between the upper bound of $\sqrt{Q \cdot S/N}$ and the $Q \cdot S/N$ advantage obtained by known attacks. Moreover, the bound's proof assumed an unproven combinatorial conjecture.

In this paper, we prove a tight upper bound (up to poly-logarithmic factors) of $O(\log Q \cdot Q \cdot S/N)$ on the adversary's advantage in the streaming distinguishing problem. The proof does not require a conjecture and is based on a reduction from communication complexity to streaming.

Keywords: Streaming algorithm, time-memory tradeoff, switching lemma, mode of operations, communication complexity.

1 Introduction

A classical result in cryptography asserts that an adversary attempting to distinguish a random permutation from a random function with an image size of N using Q queries has advantage bounded by about Q^2/N over a coin toss [2, 11, 12]. This bound serves as a switching lemma which has important implications in establishing the security of various cryptographic constructions. For example, the security of several modes of operation (such as counter-mode) is proved up to the birthday bound of $Q = \sqrt{N}$ by first idealizing the underlying block cipher as a random permutation and then replacing it with a random function using the switching lemma.[1]

A limitation of the switching lemma is that it only bounds the advantage of the adversary as a function of the number of queries, whereas in practice, the

[1] For the sake of brevity, in this paper we use the term "switching lemma" to refer to a particular type of lemma that allows to switch between a random permutation and a random function.

adversary could have constraints on additional resources, notably on memory. At the same time, given $Q \approx \sqrt{N}$ unrestricted queries to the underlying primitive, it is possible to distinguish a random function from a random permutation with constant advantage using a negligible amount of $O(\log N)$ bits of memory by applying a "memory-less" cycle detection algorithm such as Floyd's algorithm [15] (or its variants, e.g., [5, 19]).

Streaming Indistinguishability Cycle detection algorithms are inapplicable when only given access to a stream of data produced by arbitrary queries to the underlying primitive which are not under the adversary's control. The streaming indistinguishability model was introduced in the context of cryptography by Jaeger and Tessaro at EUROCRYPT 2019 [13] (related models were introduced in earlier works such as [6, 18]). The authors considered an adversary (i.e. a randomized algorithm) with memory size of S bits and access to a stream of Q elements drawn from either a random permutation or from a random function with an image size of N. The main technical result of [13] is an adaptation of the switching lemma between a random permutation and random function to the streaming model. The streaming switching lemma asserts that the adversary's advantage is bounded by $\sqrt{Q \cdot S/N}$ as long as the queries to the underlying primitive are not repeated. The proof of the bound is based on tools from information theory and relies on a combinatorial conjecture regarding hypergraphs. We refer the reader to [13] for more details.

The main applications of the switching lemma described in [13] deal with cryptanalysis of modes of operations. Such modes are typically secure up to the birthday bound against adversaries with unbounded memory, yet [13] shows that they become more secure against memory-bounded adversaries. For example, in AES-based randomized counter-mode, message m_i is encrypted as $r_i, c_i = \mathrm{AES}_K(r_i) \oplus m_i$, where r_i is a random 128-bit string. The best known distinguishing attack simply awaits a collision $r_i = r_j$ for $i \neq j$, in which case $c_i \oplus c_j = m_i \oplus m_j$. This attack stores the r_i's and requires memory of about $\sqrt{N} = 2^{64}$ to find a collision with constant probability. Let us now assume that the memory is limited to storing only $S' \ll 2^{64}$ values (where $S' \approx S \cdot \log N$, as storing S' elements requires about $S \cdot \log N$ bits). In this case, the probability of observing a collision with a stored element (i.e., the distinguishing advantage) is roughly $Q \cdot S'/N \approx Q \cdot S/N$ (ignoring a logarithmic factor in N). Hence, such a collision is likely to occur only after observing about $Q \approx N/S \gg 2^{64}$ elements.

Jaeger and Tessaro used their streaming switching lemma to show that the simple attack on randomized counter-mode describe above is optimal up to a factor of $O(\log N)$, if we require a constant advantage. The proof applies the streaming switching lemma to replace the random r_i's with random non-repeating ones and further replaces AES with a truly random permutation (assuming it is a PRP). Finally, it applies the streaming switching lemma again to replace the permutation with a random function, completely masking the messages. More details and additional applications are described in [13]. We further mention that attacks against counter-mode and other modes of operation have been shown to

be meaningful in practice (refer to [3] for a recent example), giving an additional motivation to understand their limitations.

The streaming switching lemma of [13] is very useful, but has two limitations. First, it is based on an unproven combinatorial conjecture. Second, when $Q \cdot S \ll N$, there is a gap between the advantage upper bound $\sqrt{Q \cdot S/N}$ of the lemma and the $Q \cdot S/N$ advantage of the simple attack described above. In fact, it is easy to see that the bound $\sqrt{Q \cdot S/N}$ is not tight when $Q \cdot S \ll N$ and $S \approx Q$, as it evaluates to $Q/\sqrt{N}$. On the other hand, the true optimal advantage is Q^2/N, as obtained by the original switching lemma (since for $S \approx Q$, the adversary can store all the elements in the stream).

In order to demonstrate the importance of this gap, let us assume that for $N = 2^{128}$ the adversary has memory limited to storing $S = 2^{40}$ elements, and obtains a stream of $Q = 2^{64}$ elements. Jaeger and Tessaro's result upper bounds the adversary's advantage by about $\sqrt{2^{64+40-128}} = 2^{-12}$. On the other hand, the distinguishing advantage of the attack described above is $2^{64+40-128} = 2^{-24}$, which is significantly lower.

Our Results In this paper, we overcome the two limitations of Jaeger and Tessaro's result. More specifically, we derive a streaming switching lemma which bounds the adversary's advantage by $O(\log Q \cdot Q \cdot S/N)$ via an alternative proof which it is not based on any conjecture. This matches the advantage of the simple distinguishing attack described above (up to poly-logarithmic factors in N), hence we resolve the streaming indistinguishability problem unconditionally.[2] Note that if we plug $S = Q$ into our bound, we obtain the original switching lemma (up to poly-logarithmic factors). Hence, our bound can also be viewed as a natural generalization of the original switching lemma to the case that the adversary cannot store all the Q elements of the stream (i.e. $S \ll Q$).

Finally, we extend the streaming switching lemma to show that the advantage of an adversary with S bits of memory that is allowed P passes over a stream of Q elements (drawn from a random permutation or a random function) is bounded by $O(\log Q \cdot Q \cdot S \cdot P/N)$. If we combine the multi-pass bound with the original switching lemma, we obtain the bound of about $\min\{\log Q \cdot Q \cdot S \cdot P/N, Q^2/N\}$, which is tight up to poly-logarithmic factors in N.

To understand the significance of our multi-pass bound, observe that for a fixed value of S, the P-pass streaming bound depends only on the total number of queries, $Q \cdot P$ (ignoring the small factor of $\log Q$). This essentially implies that repeating Q distinct queries P times does not give a P-pass algorithm an advantage over a single-pass algorithm that issues $Q \cdot P$ distinct queries. In contrast, in the non-streaming model repeating queries in an adaptive way has a big advantage, as cycle detection algorithms significantly beat the P-pass bound (obtaining constant advantage for $S = O(\log N)$ and $\sqrt{N}$ queries).

[2] We note, however, that Jaeger and Tessaro's result is superior to ours by a factor of $O(\log Q)$ when $S \cdot Q \approx N$.

Our Techniques The main novelty of the proof of our switching lemma is a reduction from communication complexity to streaming which is tailored to our specific cryptographic setting. Although it is simple, this reduction is somewhat non-trivial and allows us to apply strong bounds in communication complexity to the problem. This proof naturally extends to multi-pass adversaries. On the other hand, it seems challenging to extend the proof of [13] to multi-pass adversaries, where queries to the underlying primitive are repeated. This further demonstrates that our proof technique may be of independent interest.

Paper Organization The rest of the paper is organized as follows. We give a technical overview of the proof in Section 2 and describe preliminaries in Section 3. In Section 4 we prove our main streaming switching lemma for single-pass algorithms, while our proof of the multi-pass variant in given in Section 5. Finally, we conclude the paper in Section 6.

2 Technical Overview

We consider an algorithm with S bits of memory that processes a stream of $Q \leq N$ elements from $[N] = \{1, 2, \ldots, N\}$, element by element. The goal of the algorithm is to decide whether the stream is drawn from a random permutation (i.e., the elements are drawn uniformly without replacement), or from a random function (i.e., the elements are drawn uniformly with replacement).

In [13] Jaeger and Tessaro approached the problem by considering the sequences of states maintained by the adversary for the two stream distributions, claiming that they remain statistically close. Roughly speaking, the proof required a conjecture because of the difficulty in analyzing all possible adversarial strategies.

In the rest of this section, we give an overview of our proof, which (unlike Jaeger and Tessaro's proof) does not directly analyze the states maintained by the adversary. For the sake of simplicity, in this overview we only consider the range where $Q \cdot S \approx N$, and aim to show that the distinguishing advantage of the algorithm (compared to a random guess) is negligible as long as $Q \ll N/S$.

2.1 An Initial Approach

We start by informally outlining an initial approach that does not give the desired bound, but motivates the alternative approach that follows. We denote a stream drawn from a random permutation by $x_1, \ldots, x_Q$ and a stream drawn from a random function by $\hat{x}_1, \ldots, \hat{x}_Q$. One way to try and obtain the bound is to use a hybrid argument by defining intermediate stream distributions, which give rise to Q distinguishing games. The i'th game involves distinguishing between the stream distributions

$$x_1, \ldots, x_{Q-i}, \hat{x}_{Q-i+1}, \ldots, \hat{x}_Q \text{ and } x_1, \ldots, x_{Q-i-1}, \hat{x}_{Q-i}, \ldots, \hat{x}_Q,$$

which is equivalent to distinguishing between

$$x_1, \ldots, x_{Q-i} \text{ and } x_1, \ldots, x_{Q-i-1}, \hat{x}_{Q-i}.$$

Namely, the goal is to determine whether the last element already appears in the stream or not. In fact, even if the last element is chosen uniformly, it will not appear in the stream with probability $1 - (Q - i - 1)/N$. Hence, we can condition on the event that $\hat{x}_{Q-i}$ appears in the stream. As a result, the distinguishing advantage of any algorithm can be approximately bounded by $\alpha \cdot (Q - i - 1)/N$, where α is the advantage of the algorithm in distinguishing $x_1, \ldots, x_{Q-i}$ and $x_1, \ldots, x_{Q-i-1}, \hat{x}_{Q-i}$, where $\hat{x}_{Q-i}$ is drawn uniformly from the first $Q - i - 1$ elements of the stream.

A standard approach for obtaining such a bound on streaming algorithms is via a reduction from communication complexity. In our case, we consider a 2-player game between $\mathcal{A}$ and $\mathcal{B}$. Player $\mathcal{A}$ obtains $x_1, \ldots, x_{Q-i-1}$ and $\mathcal{B}$ obtains either x_{Q-i} or $\hat{x}_{Q-i}$ and the goal of $\mathcal{A}, \mathcal{B}$ is to distinguish between the cases with minimal one-way communication between $\mathcal{A}$ and $\mathcal{B}$. In the reduction from communication complexity to streaming, $\mathcal{A}$ simulates the streaming algorithm on its input, sends its state to $\mathcal{B}$, which continues the simulation of the streaming algorithm and outputs its result. Thus, any streaming algorithm with memory S yields a communication protocol with communication cost of S and the same distinguishing advantage. Therefore, an upper bound on the distinguishing advantage of $\mathcal{A}, \mathcal{B}$ yields a bound on the distinguishing advantage of the streaming algorithm.

Remark 1. $\mathcal{A}$ and $\mathcal{B}$ are unrestricted computationally, which is not a problem since we are considering distinguishing upper bounds. However, in order for the reduction from communication complexity to the streaming distinguishability game to be useful, it should have the property that for both stream distributions considered in the game, each player receives an input (partial stream) drawn from the same marginal distribution (otherwise, a player could trivially distinguish between the two distributions with no communication).

Returning to the communication complexity problem defined above, we observe that it is closely related to the *index* problem, which is a well-known problem with strong lower bounds on the communication cost required to achieve constant advantage [16]. Unfortunately, even these strong bounds are insufficient to prove the bound we require on the streaming algorithm. In order to demonstrate this, consider the following protocol: $\mathcal{A}$ hashes each element $x_1, \ldots, x_{Q-i-1}$ to a single bit, computes the majority among all bits, and sends the majority bit to $\mathcal{B}$. Then, $\mathcal{B}$ hashes its element and outputs 1 if and only if the hash is equal to the majority. Simple calculation shows that the advantage of $\mathcal{A}, \mathcal{B}$ in distinguishing between the streams is about $\alpha = 1/\sqrt{Q - i - 1}$. This implies that using this method cannot give a better upper bound than $1/\sqrt{Q - i - 1} \cdot (Q - i - 1)/N$ on the advantage of a streaming algorithm with memory $S = 1$ in distinguishing between neighboring stream distributions. If we sum over the advantages of the

first $Q-1$ games (the advantage is 0 in the last game), we obtain

$$\sum_{i=0}^{Q-2} \frac{1}{\sqrt{Q-i-1}} \cdot \frac{Q-i-1}{N} = \sum_{i=0}^{Q-2} \frac{\sqrt{Q-i-1}}{N} = \Omega\left(\frac{Q^{3/2}}{N}\right),$$

which is already $\Omega(1)$ for $Q = N^{2/3}$. On the other hand, our goal is to show that if $S = 1$ and the distinguishing advantage is $\Omega(1)$, then $Q \approx N$.

2.2 The Improved Approach

The reason that the initial attempt above fails to prove the required bound is that it uses too many intermediate hybrid distributions, and the sum of the advantages over all Q games results in a significant loss in the bound. We discuss two alternative approaches to overcome the loss. The first alternative approach is to try and avoid the straightforward sum of advantages by using more advanced techniques developed in the area of provable security for the purpose of obtaining tight bounds (e.g., the chi-squared method proposed in [8]). However, such techniques do not directly apply to the streaming model where the adversary no longer has access to answers of its previous queries. Moreover, it seems challenging to extend such techniques to the multi-pass setting in order to handle the dependencies between repeated queries to the underlying primitive. In this paper, we use a completely different approach by reconsidering our definition of intermediate hybrid distributions that lead from a stream produced by random permutation to a stream produced by a random function.

The First Intermediate Hybrid Distribution We start by defining the first distinguishing game between $x_1, \ldots, x_Q$ (a stream drawn from a random permutation) and a second stream drawn from a carefully chosen hybrid distribution. Our goals in defining the game are: (1) in order to minimize the number of hybrid distributions, the first intermediate hybrid should break the dependency among a maximal number of elements of $x_1, \ldots, x_Q$, and (2) the distinguishability bound for the two stream distributions should be derived from a 2-player communication game in which the marginal distributions of the inputs given to each player are identical. Note that these two conditions are somewhat conflicting, as (2) restricts the inputs of each $\mathcal{A}$ and $\mathcal{B}$ for both stream distributions to contain no repetitions.

We define our stream distributions using the notation $x_1^1, \ldots, x_{Q/2}^1, y_1^1, \ldots, y_{Q/2}^1$, where each of $x_1^1, \ldots, x_{Q/2}^1$ and $y_1^1, \ldots, y_{Q/2}^1$ is a stream drawn from a random permutation, such that the streams are either drawn from the same permutation (this corresponds to the stream $x_1, \ldots, x_Q$), or drawn from independent permutations (which corresponds to the first intermediate hybrid). We then define the corresponding 2-player communication problem (which we call the *permutation-dependence* problem), where $\mathcal{A}$ and $\mathcal{B}$ obtain $x_1^1, \ldots, x_{Q/2}^1$ and $y_1^1, \ldots, y_{Q/2}^1$, respectively, and try to decide with minimal communication whether their inputs are drawn from the same or from independent permutations.

To complete the distinguishability upper bound proof for the streaming game, we prove an upper bound on the distinguishing advantage of $\mathcal{A}$ and $\mathcal{B}$ in the permutation-dependence problem. The proof is by a reduction from the *set-disjointness* problem, which is a canonical 2-player problem in communication complexity [1, 14, 20], where the input of each player is a set and their goal is to determine whether their sets intersect, or are disjoint.[3] Interestingly, in order to obtain our optimal bound for the range $Q \cdot S \ll N$, we have to use recent results for set-disjointness [4, 10] which improve upon the classical bounds for low success probabilities that are particularly relevant for cryptography.

The Remaining Hybrid Distributions We define the second intermediate hybrid using a similar approach by breaking the two halves of the stream into quarters

$$x_1^1, \ldots, x_{Q/4}^1, y_1^1, \ldots, y_{Q/4}^1, x_1^2, \ldots, x_{Q/4}^2, y_1^2, \ldots, y_{Q/4}^2,$$

which corresponds to a stream consisting of either 2 streams of length $Q/2$ drawn from 2 independent permutations (the first intermediate hybrid), or 4 streams of length $Q/4$ drawn from 4 independent permutations (the second intermediate hybrid). In the corresponding (generalized) permutation-dependence problem, each player obtains $Q/2$ elements drawn from 2 independent permutations, while in the reduction to streaming, $\mathcal{A}'s$ input corresponds to the x values, while B's input corresponds to the y values of the stream. The remaining hybrid distributions are defined in a similar structure that resembles a binary tree. The i'th distinguishing game corresponds to a stream consisting of elements drawn from either 2^{i-1} or 2^i independent permutations, and in the permutation-dependence problem, we give $\mathcal{A}$ and $\mathcal{B}$ interleaving streams, each consisting of $Q/2^i$ elements. Overall, we have $\log Q$ games (and $\log Q - 2$ intermediate hybrids), where the final game corresponds to distinguishing between a stream consisting of $Q/2$ element pairs drawn from independent permutations, and Q elements drawn from a random function.[4]

Note that in all distinguishing games except for the first, the inputs of $\mathcal{A}$ and $\mathcal{B}$ in the corresponding permutation-dependence problem consist of interleaving (non-continuous) streams. This implies that the state of the streaming algorithm has to be communicated several times in the reduction to streaming and results in a loss in the upper bound compared to the first game. On the other hand, the reduction from set-disjointness to the permutation-dependence problem on the shorter independent streams gives a better bound compared to the first game, compensating for the loss. Overall, the loss of $\log Q$ in our bound $O(\log Q \cdot Q \cdot S/N)$ is attributed to the $\log Q$ games, each giving a similar bound of $O(Q \cdot S/N)$.

[3] In fact, the reduction is from the *unique-disjointness* problem which is a variant of set-disjointness with the promise the if the sets of the players intersect, the intersection size is 1.

[4] A hybrid argument on a binary tree is also used to prove the security of the classical pseudo-random function construction by Goldreich et al. [9]. However, the resemblance is superficial, as in [9] the construction itself is a binary tree, whereas in our case, we build it artificially only in the proof.

3 Preliminaries

Unless stated explicitly, all parameters considered in this paper are positive integers. We define $[N] = \{1, 2, \ldots, N\}$ and let $[N]^K = \underbrace{[N] \times [N] \times \ldots \times [N]}_{K}$.

Given bit strings x and y, we denote their concatenation by $x\|y$. For a positive integer K, we denote by $x^{(K)}$ the string $\underbrace{x\|x\ldots\|x}_{K}$, obtained by K repetitions of x. We denote by $HW(x)$ the Hamming weight of x.

Given a bit string $a \in \{0,1\}^N$ such that $HW(a) = K$, we can treat it as an incidence vector of a set $\{x_1, x_2, \ldots, x_K\}$ such that $x_i \in [N]$ and $a[x_i] = 1$ for $i \in [K]$. We define $SEQ : \{0,1\}^N \to [N]^K$ as the sequence $SEQ(a) = x_1, x_2, \ldots, x_K$ (which includes the elements indicated by a in lexicographical order). Given incidence vectors $a \in \{0,1\}^N$ and $b \in \{0,1\}^N$, let $a \cap b$ denote the intersection of these sets, and $|a \cap b|$ the size of the intersection.

Given a distribution $\mathcal{X}$ on strings with finite support, we write $x \xleftarrow{\$} \mathcal{X}$ to denote a random variable x chosen from $\mathcal{X}$. We write $x \sim \mathcal{X}$ if x is a random variable that is distributed as $\mathcal{X}$.

Distinguishing between Streams We define our model for a randomized algorithm whose goal is to distinguish between streams. The model is similar to the one defined in [13], although we use slightly different notation.

For some parameters N, K, let $\mathcal{X}$ be some distribution over $[N]^K$. We denote by $\mathrm{O}(\mathcal{X})$ an oracle that samples $x_1, x_2, \ldots, x_K$ from $\mathcal{X}$. The oracle receives up to K queries and answers query number i by x_i. Note that once the oracle outputs x_i, it is not output again. This implies that an algorithm $\mathcal{A}$ that interacts with $\mathrm{O}(\mathcal{X})$ receives $x_1, x_2, \ldots, x_K$ as a stream, i.e., if $\mathcal{A}$ requires access to x_i after issuing query i, it has to store x_i in memory in some representation.

We denote by $\mathcal{A}^{\mathrm{O}(\mathcal{X})}$ a randomized algorithm with oracle access to $\mathrm{O}(\mathcal{X})$ and by $\mathcal{A}^{\mathrm{O}(\mathcal{X})} \Rightarrow b$ the event that the algorithm outputs the bit $b \in \{0,1\}$.

We say that an algorithm $\mathcal{A}$ is S-bounded, if the size of each state maintained by $\mathcal{A}$ during any execution is upper bounded by S bits.

Let $\mathcal{X}$ and $\mathcal{Y}$ be two distributions over $[N]^K$. The streaming distinguishing advantage of an algorithm $\mathcal{A}$ between $\mathcal{X}$ and $\mathcal{Y}$ is defined as

$$\mathrm{Adv}^{\mathrm{STR}}_{\mathcal{X},\mathcal{Y}}(\mathcal{A}) = \left|\Pr[\mathcal{A}^{\mathrm{O}(\mathcal{X})} \Rightarrow 1] - \Pr[\mathcal{A}^{\mathrm{O}(\mathcal{Y})} \Rightarrow 1]\right|.$$

We further define the optimal advantage for an S-bounded algorithm as

$$\mathrm{Opt}^{\mathrm{STR}}_{\mathcal{X},\mathcal{Y}}(S) = \limsup_{\mathcal{A}} \{\mathrm{Adv}^{\mathrm{STR}}_{\mathcal{X},\mathcal{Y}}(\mathcal{A}) \mid \mathcal{A} \text{ is } S - \text{bounded}\}.$$

Sampling with and without Replacement For a parameter $0 < K \leq N$, let $\mathcal{D}^K_N$ be the distribution over $[N]^K$ that is defined by a sampling procedure which uniformly draws K elements from $[N]$ without replacement.

For parameters $0 < K \leq N$ and $R > 0$, let $\mathcal{D}_N^{K \times R}$ be the distribution over $[N]^{K \cdot R}$ that is composed of R independent copies of $\mathcal{D}_N^K$.

Note that sampling from $\mathcal{D}_N^{1 \times K}$ is equivalent to choosing K items from $[N]$ uniformly with replacement (i.e., from a random function), while sampling from $\mathcal{D}_N^K$ is equivalent to choosing K items from $[N]$ uniformly without replacement (i.e., from a random permutation).

The original switching lemma between a random permutation and a random function [2, 11, 12] asserts that any algorithm that issues Q queries to the underlying primitive has distinguishing advantage bounded by $Q^2/2N$. This bound obviously holds in the (more restricted) streaming model.

Theorem 1 (switching lemma [2, 11, 12]). *For any S and $Q \leq N$,*

$$\mathrm{Opt}^{\mathrm{STR}}_{\mathcal{D}_N^Q, \mathcal{D}_N^{1 \times Q}}(S) \leq \frac{Q^2}{2N}.$$

The Set-Disjointness and Unique-Disjointness Problems

The set-disjointness function $DISJ : \{0,1\}^N \times \{0,1\}^N \to \{0,1\}$ is defined as

$$DISJ(a,b) = \begin{cases} 0, & \text{there exists } i \in [N] \text{ for which } a[i] = b[i] = 1 \\ 1, & \text{otherwise.} \end{cases}$$

We can view a and b as subsets of $[N]$, encoded as incidence vectors, and then $DISJ(a,b) = 1$ if a and b are disjoint.

The *set-disjointness problem* (or *disjointness* in short) is a classical problem in communication complexity.[5] It is a 2-player game between $\mathcal{A}$ and $\mathcal{B}$ that run a protocol Π. In an instance of disjointness $\mathcal{A}$ receives $a \in \{0,1\}^N$, $\mathcal{B}$ receives $b \in \{0,1\}^N$ and their goal is to output $DISJ(a,b)$ with minimal communication in the worst case. Namely, the communication cost of Π is defined as the maximal number of bits communicated among all possible protocol executions.

We consider a variant of the disjointness problem called *unique-disjointness*, which is identical to disjointness, but with the promise that in a 0-instance, there exists a *single* index $i \in [N]$ for which $a[i] = b[i] = 1$. We denote the corresponding function by $UDISJ$, where we define $UDISJ(a,b) = \perp$ if a, b do not satisfy the required promise. We will be interested in a public-coin randomized variant of unique-disjointness in which $\mathcal{A}, \mathcal{B}$ have access to a shared random string that is independent of their inputs.

We denote the output of the protocol Π on inputs a, b as $UDISJ_\Pi(a,b)$. Note that it is a random variable that depends on the shared randomness of $\mathcal{A}, \mathcal{B}$. Disjointness and its variants are worst case problems. This motivates the

[5] For a (slightly outdated) survey on set-disjointness, refer to [7].

following notation for the error and advantage of the protocol.[6]

$$\begin{aligned}
\mathrm{Err}_N^{\mathrm{UDISJ0}}(\Pi) &= \max_{a,b}\{\Pr[UDISJ_\Pi(a,b) \neq 0 \mid UDISJ(a,b) = 0]\},\\
\mathrm{Err}_N^{\mathrm{UDISJ1}}(\Pi) &= \max_{a,b}\{\Pr[UDISJ_\Pi(a,b) \neq 1 \mid UDISJ(a,b) = 1]\},\\
\mathrm{Err}_N^{\mathrm{UDISJ}}(\Pi) &= \max\{\mathrm{Err}_N^{\mathrm{UDISJ0}}(\Pi), \mathrm{Err}_N^{\mathrm{UDISJ1}}(\Pi)\},\\
\mathrm{Adv}_N^{\mathrm{UDISJ}}(\Pi) &= \left|1 - \mathrm{Err}_N^{\mathrm{UDISJ1}}(\Pi) - \mathrm{Err}_N^{\mathrm{UDISJ0}}(\Pi)\right|.
\end{aligned}$$

The following is a classical result in communication complexity.

Theorem 2 ([1, 14, 20, adapted]). *Any public-coin randomized protocol Π that solves unique-disjointness on all inputs $a, b \in \{0,1\}^N \times \{0,1\}^N$ such that $UDISJ(a,b) \in \{0,1\}$ with error probability $\mathrm{Err}_N^{\mathrm{UDISJ}}(\Pi) \leq 1/3$, uses $\Omega(N)$ bits of communication in the worst case.*

Therefore, it is not possible to do much better than the trivial protocol in which $\mathcal{A}$ sends $\mathcal{B}$ its entire input a, and $\mathcal{B}$ outputs $UDISJ(a,b)$.

When analyzing the advantage γ of a protocol with communication cost of $o(N)$, we can repeat it and amplify its advantage using a majority vote to obtain an error probability of at most $1/3$. By applying a Chernoff bound and using Theorem 2, we can lower bound the communication cost required to achieve advantage of γ by $\Omega(\gamma^2 N)$. If we use this bound for the purpose of obtaining a streaming switching lemma, we get a result which is similar to the $\sqrt{Q \cdot S/N}$ bound of [13]. However, relatively recent results [4, 10] prove a much stronger lower bound of $\Omega(\gamma N)$ on the communication cost by a more careful analysis. This stronger bound (summarized in the theorem below) will allow us to prove an improved streaming switching lemma.

Theorem 3 ([10, Theorem 1.5, adapted]). *For a public-coin randomized protocol for unique-disjointness Π, denote*

$$\alpha(N) = 1 - \mathrm{Err}_N^{\mathrm{UDISJ1}}(\Pi) \text{ and } \beta(N) = \mathrm{Err}_N^{\mathrm{UDISJ0}}(\Pi).$$

There exist constants $0 < M_1 < 1$ and $M_2 > 0$ such that for all $\alpha(N) > \beta(N)$ that satisfy

$$\log(1/\alpha) \leq M_1 \cdot N \cdot (1 - \beta/\alpha),$$

the communication cost of Π is at least $M_2 \cdot N \cdot (1 - \beta/\alpha)$ bits in the worst case.

Remark 2. The theorem is stated in [10] for the set-disjointness problem, rather than for unique-disjointness. However, the proof actually considers unique-disjointness. Since set-disjointness is a worst-case problem, the lower bound on unique-disjointness also applies to set-disjointness.

Remark 3. We could have also used Theorem 2.2 of [4], and adapted it to our purposes (we need a public-coin randomized variant of this theorem).

[6] Our notation for disjointness is consistent with the rest of the paper, yet it differs from standard notation used in communication complexity.

It would be useful for us to bound the communication cost of the protocol using it advantage.

Corollary 1. *For a public-coin randomized protocol for unique-disjointness Π, denote*

$$\gamma(N) = \mathrm{Adv}_N^{\mathrm{UDISJ}}(\Pi).$$

Then, there exist constants $0 < M_1 < 1$ and $M_2 > 0$ such that for all $\gamma(N)$ that satisfy

$$\gamma(N) \geq \log N/(M_1 \cdot N),$$

the communication cost of Π is at least $M_2 \cdot N \cdot \gamma$ bits in the worst case.

Proof. We assume that the conditions of Corollary 1 hold and show that the conditions of Theorem 3 hold. We use the definition of α and β and constants M_1, M_2 in Theorem 3 (M_1, M_2 are the same constants in Corollary 1). We have $\alpha - \beta = \gamma \geq \log N/(M_1 \cdot N) > 0$. In addition, $\alpha \geq \gamma \geq \log N/(M_1 \cdot N) > 1/N$, hence $\log(1/\alpha) < \log N$. Therefore,

$$M_1 \cdot N \cdot (1 - \beta/\alpha) = M_1 \cdot N \cdot (\gamma/\alpha) \geq$$
$$M_1 \cdot N \cdot \gamma \geq M_1 \cdot N \cdot \log N/(M_1 \cdot N) = \log N \geq \log(1/\alpha).$$

Thus, we can apply Theorem 3 and conclude that the cost of Π is at least

$$M_2 \cdot N \cdot (1 - \beta/\alpha) = M_2 \cdot N \cdot (\gamma/\alpha) \geq M_2 \cdot N \cdot \gamma$$

bits in the worst case. ■

4 The Streaming Switching Lemma

Our main theorem is stated below. We refer to it as a "streaming switching lemma" (for the sake of compatibility with previous results).

Theorem 4 (streaming switching lemma). *There exists a constant $0 < M < 1$ such that any S-bounded randomized algorithm $\mathcal{A}$ with access to a stream containing $\log N \leq Q \leq N/3$ elements drawn from $[N]$ via either a random permutation or a random function has a distinguishing advantage bounded by*

$$\mathrm{Adv}^{\mathrm{STR}}_{\mathcal{D}_N^Q, \mathcal{D}_N^{1\times Q}}(\mathcal{A}) \leq \mathrm{Opt}^{\mathrm{STR}}_{\mathcal{D}_N^Q, \mathcal{D}_N^{1\times Q}}(S) \leq \frac{\lceil \log Q \rceil}{M} \cdot \frac{Q \cdot S}{N} < \frac{\log N}{M} \cdot \frac{Q \cdot S}{N}.$$

Remark 4. When $Q < \log N$, we can use the original switching lemma (Theorem 1) to bound the advantage by $Q^2/(2N) \leq \log^2 N/(2N)$.

Theorem 4 follows from the lemma below.

Lemma 1. *There exists a constant $0 < M < 1$ such that for any $K \leq N/3$ and $S \cdot R \geq \log N$,*

$$\mathrm{Opt}^{\mathrm{STR}}_{\mathcal{D}_N^{2K\times R}, \mathcal{D}_N^{K\times 2R}}(S) \leq \frac{1}{M} \cdot \frac{S \cdot R \cdot K}{N}.$$

Proof (of Theorem 4). Let $\mathcal{A}$ be an S-bounded algorithm and let M be the constant (implied by Lemma 1) such that $\mathrm{Adv}^{\mathrm{STR}}_{\mathcal{D}_N^{2K\times R},\mathcal{D}_N^{K\times 2R}}(\mathcal{A}) \leq \frac{1}{M}\cdot\frac{S\cdot R\cdot K}{N}$.

Assume that $\log N \leq Q \leq N/3$, and let $q' = \lceil \log Q \rceil$ and $Q' = 2^{q'}$ (note that $Q \leq Q' \leq 2Q$). We have

$$\begin{aligned}
\mathrm{Adv}^{\mathrm{STR}}_{\mathcal{D}_N^{Q},\mathcal{D}_N^{1\times Q}}(\mathcal{A}) \leq \mathrm{Adv}^{\mathrm{STR}}_{\mathcal{D}_N^{Q'},\mathcal{D}_N^{1\times Q'}}(\mathcal{A}) = \\
\left|\Pr[\mathcal{A}^{\mathrm{O}(\mathcal{D}_N^{Q'})} \Rightarrow 1] - \Pr[\mathcal{A}^{\mathrm{O}(\mathcal{D}_N^{1\times Q'})} \Rightarrow 1]\right| = \\
\Big|\sum_{i=0}^{q'-1}\left(\Pr[\mathcal{A}^{\mathrm{O}(\mathcal{D}_N^{Q'/2^i\times 2^i})} \Rightarrow 1] - \Pr[\mathcal{A}^{\mathrm{O}(\mathcal{D}_N^{Q'/2^{i+1}\times 2^{i+1}})} \Rightarrow 1]\right)\Big| \leq \\
\sum_{i=0}^{q'-1}\left|\left(\Pr[\mathcal{A}^{\mathrm{O}(\mathcal{D}_N^{Q'/2^i\times 2^i})} \Rightarrow 1] - \Pr[\mathcal{A}^{\mathrm{O}(\mathcal{D}_N^{Q'/2^{i+1}\times 2^{i+1}})} \Rightarrow 1]\right)\right| = \\
\sum_{i=0}^{q'-1}\mathrm{Adv}^{\mathrm{STR}}_{\mathcal{D}_N^{Q'/2^i\times 2^i},\mathcal{D}_N^{Q'/2^{i+1}\times 2^{i+1}}}(\mathcal{A}) \leq \\
\frac{q'}{M}\cdot\frac{Q'\cdot S}{2N} \leq \frac{\lceil\log Q\rceil}{M}\cdot\frac{Q\cdot S}{N},
\end{aligned}$$

where the penultimate inequality follows from Lemma 1. ■

4.1 Reduction from Communication Complexity to Streaming

We now define the permutation-dependence problem and summarize the outcome of the reduction from this problem to streaming in Proposition 1. We then state a lower bound on the communication complexity cost of the permutation-dependence problem in Proposition 2 (which is proved in Section 4.2), and use it to prove Lemma 1.

The Permutation-Dependence Problem *Permutation-dependence* is a 2-player game between $\mathcal{A}$ and $\mathcal{B}$ that run a protocol Π. For parameters K, R such that K is even and $K \cdot R \leq N$, we choose the $K \cdot R$ elements

$$x_1^1,\ldots,x_{K/2}^1,y_1^1,\ldots,y_{K/2}^1,x_1^2,\ldots,x_{K/2}^2,y_1^2,\ldots,y_{K/2}^2,\ldots,x_1^R,\ldots,x_{K/2}^R,y_1^R,\ldots,y_{K/2}^R,$$

from either $\mathcal{D}_N^{K\times R}$, or from $\mathcal{D}_N^{K/2\times 2R}$. We give

$$x_1^1,\ldots,x_{K/2}^1,x_1^2,\ldots,x_{K/2}^2,\ldots,x_1^R,\ldots,x_{K/2}^R$$

to $\mathcal{A}$ and

$$y_1^1,\ldots,y_{K/2}^1,y_1^2,\ldots,y_{K/2}^2,\ldots,y_1^R,\ldots,y_{K/2}^R$$

to $\mathcal{B}$. Note that regardless of the distribution from which the $K \cdot R$ elements are chosen, the input to each player is taken from the (marginal) distribution

$\mathcal{D}_N^{K/2\times R}$. However, the inputs are either dependent (chosen from $\mathcal{D}_N^{K\times R}$) or independent (chosen from $\mathcal{D}_N^{K/2\times 2R}$) and the goal of the players is to distinguish between these cases.

After receiving their inputs x, y, players $\mathcal{A}, \mathcal{B}$ run communication protocol Π and then one of the players outputs a bit which is the output of the protocol, denoted by $PDEP_\Pi(x, y)$. We say that Π has communication cost C if $\mathcal{A}, \mathcal{B}$ communicate at most C bits in all possible protocol executions. Similarly to the disjointness problem, we will be interested in public-coin randomized protocols for permutation-dependence.

Since it is a distributional communication complexity problem, we define the following notation for permutation-dependence:

$$\begin{aligned}
\mathrm{Err}^{\mathrm{PDEP0}}_{N,K,R}(\Pi) &= \Pr[PDEP_\Pi(x,y) = 1 \mid x, y \xleftarrow{\$} \mathcal{D}_N^{K/2\times 2R}],\\
\mathrm{Err}^{\mathrm{PDEP1}}_{N,K,R}(\Pi) &= \Pr[PDEP_\Pi(x,y) = 0 \mid x, y \xleftarrow{\$} \mathcal{D}_N^{K\times R}],\\
\mathrm{Adv}^{\mathrm{PDEP}}_{N,K,R}(\Pi) &= \left|1 - \mathrm{Err}^{\mathrm{PDEP1}}_{N,K,R}(\Pi) - \mathrm{Err}^{\mathrm{PDEP0}}_{N,K,R}(\Pi)\right|,\\
\mathrm{Opt}^{\mathrm{PDEP}}_{N,K,R}(C) &= \limsup_{\Pi}\{\ \mathrm{Adv}^{\mathrm{PDEP}}_{N,K,R}(\Pi) \mid \Pi \text{ has communication cost } C\}.
\end{aligned}$$

The Reduction from Permutation-Dependence to Streaming The following proposition upper bounds the advantage of a (memory-bounded) streaming algorithm in distinguishing between $\mathcal{D}_N^{K\times R}$ and $\mathcal{D}_N^{K/2\times 2R}$ by the advantage of an optimal permutation-dependence protocol (with limited communication cost). It is a standard reduction from a 2-player communication protocol to streaming (for example, refer to [17]).

Proposition 1. *For any even $K \le N$,*

$$\mathrm{Opt}^{\mathrm{STR}}_{\mathcal{D}_N^{K\times R}, \mathcal{D}_N^{K/2\times 2R}}(S) \le \mathrm{Opt}^{\mathrm{PDEP}}_{N,K,R}(S \cdot R).$$

Proof. Given black-box access to an S-bounded streaming algorithm $\mathcal{A}_1$, players $\mathcal{A}$ and $\mathcal{B}$ in the permutation-dependence protocol Π run $\mathcal{A}_1$ and answer its oracle queries using their inputs: $\mathcal{A}$ answers the first batch of $K/2$ queries (using $x_1^1, \ldots, x_{K/2}^1$) and then communicates the state of $\mathcal{A}_1$ to $\mathcal{B}$ which answers the second batch of K queries (using $y_1^1, \ldots, y_{K/2}^1, y_1^2, \ldots, y_{K/2}^2$). $\mathcal{B}$ then communicates the state of $\mathcal{A}_1$ back to $\mathcal{A}$ which answers the third batch of K queries (using $x_1^2, \ldots, x_{K/2}^2, x_1^3, \ldots, x_{K/2}^3$), and so forth. Finally, $\mathcal{B}$ answers the final batch of $K/2$ queries using $y_1^R, \ldots, y_{K/2}^R$ and outputs the same answer as $\mathcal{A}_1$.

Thus, $\mathcal{A}_1$ is given oracle access to O, where either $\mathrm{O} = \mathrm{O}(\mathcal{D}_N^{K\times R})$ or $\mathrm{O} = \mathrm{O}(\mathcal{D}_N^{K/2\times 2R})$, depending on the distribution of the inputs x, y of $\mathcal{A}, \mathcal{B}$. Moreover, since $\mathcal{A}_1$ is S-bounded and its state is communicated R times, the communication cost of Π is bounded by $S \cdot R$. Therefore,

$$\mathrm{Adv}^{\mathrm{STR}}_{\mathcal{D}_N^{K\times R}, \mathcal{D}_N^{K/2\times 2R}}(\mathcal{A}_1) = \mathrm{Adv}^{\mathrm{PDEP}}_{N,K,R}(\Pi) \le \mathrm{Opt}^{\mathrm{PDEP}}_{N,K,R}(S \cdot R).$$

The proposition follows since the above inequality holds for any S-bounded algorithm $\mathcal{A}_1$. ■

Remark 5. In case $S > K/2$, a trivial reduction (where one party sends its input to the other) is more efficient than the one above. This gives

$$\mathrm{Opt}^{\mathrm{STR}}_{\mathcal{D}_N^{K\times R},\mathcal{D}_N^{K/2\times 2R}}(S) \leq \mathrm{Opt}^{\mathrm{PDEP}}_{N,K,R}(K\cdot R/2).$$

Using this observation, it is possible to obtain a limited improvement to the streaming switching lemma of Theorem 4 in case $S = N^{\omega(1)}$.

Proof of Lemma 1 In order to prove Lemma 1, we use the following proposition which bounds the advantage of any protocol Π for permutation-dependence.

Proposition 2. *There exists a constant $0 < M < 1$ such that for any R, $K \leq N/3$ and $C \geq \log N$,*

$$\mathrm{Opt}^{\mathrm{PDEP}}_{N,2K,R}(C) \leq \frac{1}{M}\cdot\frac{C\cdot K}{N}.$$

Proof (of Lemma 1). Let M be the constant implied by Proposition 2. Based on Proposition 1 and Proposition 2 we have

$$\mathrm{Opt}^{\mathrm{STR}}_{\mathcal{D}_N^{2K\times R},\mathcal{D}_N^{K\times 2R}}(S) \leq \mathrm{Opt}^{\mathrm{PDEP}}_{N,2K,R}(S\cdot R) \leq \frac{1}{M}\cdot\frac{S\cdot R\cdot K}{N}$$

■

4.2 Reduction from Unique-Disjointness to Permutation-Dependence

The proof of Proposition 2 is based on a reduction from the unique-disjointness problem to the permutation-dependence problem.

Proposition 3. *Let $K \leq N/3$ and $N' = \lfloor N/K \rfloor$. There exists a public-coin randomized local reduction, f_1, f_2, where $f_i : \{0,1\}^{N'} \to [N]^{K\cdot R}$, such that for any $a, b \in \{0,1\}^{N'} \times \{0,1\}^{N'}$,*

$$f_1(a), f_2(b) \sim \begin{cases} \mathcal{D}_N^{K\times 2R}, & \text{if } UDISJ(a,b) = 0 \\ \mathcal{D}_N^{2K\times R}, & \text{if } UDISJ(a,b) = 1. \end{cases}$$

Here, a public-coin randomized local reduction means that f_1 only depends of a and on public randomness (but not on b), and similarly, f_2 does not depend on a. In the particular case of $R = 1$, if a, b intersect at exactly 1 index, then the output of the reduction consists of two independent random permutation streams, each of K elements. On the other hand, if a, b are disjoint, then the output of the reduction consists of a single random permutation stream of $2K$ elements (that is split among the parties).

Proof. We first describe the reduction f_1, f_2 for the specific case of $R = 1$ below, as a procedure executed by two parties $\mathcal{A}, \mathcal{B}$ that do not communicate, but share a random string.

1. Given incidence vector inputs (bit arrays) $a, b \in \{0,1\}^{N'} \times \{0,1\}^{N'}$, let $S_A = a^{(K)} \| 0^{(N-N' \cdot K)}, S_B = b^{(K)} \| 0^{(N-N' \cdot K)}$. Namely, each party locally duplicates its array K times and appends zero entries such that $S_A \in \{0,1\}^N$ and $S_B \in \{0,1\}^N$.
2. Using their joint randomness, the parties sample a sequence of K indices $i_1, i_2, \ldots, i_K \xleftarrow{\$} \mathcal{D}_N^K$ (chosen from $[N]$ without replacement). The parties use the sampled indices to create new arrays: $\mathcal{A}$ defines an array $T_A \in \{0,1\}^K$, where $T_A[j] = S_A[i_j]$ for $j \in \{1, 2, \ldots, K\}$. Similarly, $\mathcal{B}$ defines $T_B \in \{0,1\}^K$, where $T_B[j] = S_B[i_j]$ for $j \in \{1, 2, \ldots, K\}$.
3. Each party locally extends its array from size K to size N such that its Hamming weight becomes K (the parties add disjoint 1 entries). More specifically, $\mathcal{A}$ computes

$$T_A^2 = T_A \| 1^{(K - HW(T_A))} \| 0^{(N - 2K + HW(T_A))},$$

   and $\mathcal{B}$ computes

$$T_B^2 = T_B \| 0^{(K)} \| 1^{(K - HW(T_B))} \| 0^{(N - 3K + HW(T_B))}.$$

4. Each party applies (the same) uniform permutation $\sigma : \{0,1\}^N \to \{0,1\}^N$ to its array of size N (σ is specified in the joint randomness),

$$T_A^3[i] = T_A^2[\sigma(i)], \text{ and } T_B^3[i] = T_B^2[\sigma(i)],$$

   for each $i \in [N]$.
5. Finally, $\mathcal{A}$ selects a uniform permutation $\sigma_1 : \{0,1\}^K \to \{0,1\}^K$ and uses it to output the elements indicated by its array T_A^3 (the 1 entries) in uniform order. $\mathcal{A}$ outputs

$$f_1(a)_i = SEQ(T_A^3)_{\sigma_1(i)}, \text{ for each } i \in [K].$$

   $\mathcal{B}$ selects a uniform permutation $\sigma_2 : \{0,1\}^K \to \{0,1\}^K$ and outputs

$$f_2(b)_i = SEQ(T_B^3)_{\sigma_2(i)}, \text{ for each } i \in [K].$$

Analysis Observe that $T_A^3 \in \{0,1\}^N$ satisfies $HW(T_A^3) = K$ and similarly $T_B^3 \in \{0,1\}^N$ satisfies $HW(T_B^3) = K$. Therefore, each party outputs a sequence of K elements.

Due to the randomization of σ (which randomizes the elements that are output by f_1, f_2) and of σ_1, σ_2 (which randomize the order of the elements output by f_1, f_2), we have the following property.

Property 1. Let $a, b \in \{0,1\}^{N'} \times \{0,1\}^{N'}$ and

$$x, y = x_1, \ldots, x_K, y_1, \ldots, y_K \in [N]^{2K},\ x', y' = x'_1, \ldots, x'_K, y'_1, \ldots, y'_K \in [N]^{2K},$$

where each K element sequence (x, y, x' and y') contains distinct elements and for some $0 \leq t \leq K$,

$$|\{x_1, \ldots, x_K\} \cap \{y_1, \ldots, y_K\}| = |\{x'_1, \ldots, x'_K\} \cap \{y'_1, \ldots, y'_K\}| = t.$$

Then,

$$\Pr[f_1(a), f_2(b) = x, y] = \Pr[f_1(a), f_2(b) = x', y'].$$

Hence, the distribution of $f_1(a), f_2(b)$ is completely determined by the distribution of the size of the intersection of the sequences $f_1(a)$ and $f_2(b)$ as sets. The intersection size is equal to $|T_A \cap T_B|$ (since $|T_A \cap T_B| = |T_A^3 \cap T_B^3|$), thus we analyze this variable below.

Observe that

$$|S_A \cap S_B| = K \cdot |a \cap b|.$$

Consider the case that $UDISJ(a, b) = 1$, or $|a \cap b| = 0$. We have $|S_A \cap S_B| = 0$ and therefore $|T_A \cap T_B| = 0$. Hence, $f_1(a)$ and $f_2(b)$ are disjoint as sets, and by Property 1, $f_1(a), f_2(b) \sim \mathcal{D}_N^{2K \times 1}$.

Otherwise, $UDISJ(a, b) = 0$, implying that $|a \cap b| = 1$ and therefore $|S_A \cap S_B| = K$. the number of options for selecting $i_1, i_2, \ldots, i_K$ in the second step such that they intersect the K common indices in S_A, S_B in exactly $0 \leq t \leq K$ places is $\binom{K}{t}\binom{N-K}{K-t}$. Since the total number of options for selecting $i_1, i_2, \ldots, i_K$ is $\binom{N}{K}$,

$$\Pr[|T_A \cap T_B| = t] = \frac{\binom{K}{t}\binom{N-K}{K-t}}{\binom{N}{K}}.$$

At the same time,

$$\Pr[\left|\{x_1, \ldots, x_K\} \cap \{y_1, \ldots, y_K\}\right| = t \mid x_1, \ldots, x_K, y_1, \ldots, y_K \xleftarrow{\$} \mathcal{D}_N^{K \times 2}] = \frac{\binom{K}{t}\binom{N-K}{K-t}}{\binom{N}{K}} = \Pr[|T_A \cap T_B| = t].$$

Hence, by Property 1, $f_1(a), f_2(b) \sim \mathcal{D}_N^{K \times 2}$ as claimed.

The generalization of the reduction for $R > 1$ and its analysis are straightforward: the parties independently generate R stream pairs $f_1^i(a), f_2^i(b)$ for $i \in \{1, \ldots, R\}$ using S_A, S_B, repeating all the steps (except the first) with fresh randomness. ■

Finally, Proposition 2 follows from Proposition 3 by applying Corollary 1.

Proof (of Proposition 2). We show that there exists a constant M such that any permutation-dependence protocol Π' with communication cost $C \geq \log N$, satisfies $\mathrm{Adv}^{\mathrm{PDEP}}_{N,2K,R}(\Pi') \leq C \cdot K/(M \cdot N)$. This proves Proposition 2.

We consider a protocol Π for unique-disjointness, where given an input $a, b \in \{0,1\}^{N'} \times \{0,1\}^{N'}$ (for $N' = \lfloor N/K \rfloor$), each party independently applies the reduction of Proposition 3 to its input using the public randomness. The parties then run the permutation-dependence protocol Π' on input $f_1(a), f_2(b)$ with

communication cost (at most) C bits in the worst case and output the same value. In short,

$$UDISJ_{\Pi}(a,b) = PDEP_{\Pi'}(f_1(a), f_2(b)).$$

The reduction of Proposition 3 implies that for every a, b such that $UDISJ(a,b) = 0$,

$$\Pr[UDISJ_{\Pi}(a,b) = 1 \mid UDISJ(a,b) = 0] =$$
$$\Pr[PDEP_{\Pi'}(f_1(a), f_2(b)) = 1 \mid UDISJ(a,b) = 0] = \mathrm{Err}^{\mathrm{PDEP0}}_{N,2K,R}(\Pi'),$$

and a similar equality holds for every a, b such that $UDISJ(a,b) = 1$. Hence

$$\mathrm{Err}^{\mathrm{UDISJ0}}_{N'}(\Pi) = \mathrm{Err}^{\mathrm{PDEP0}}_{N,2K,R}(\Pi'), \text{ and } \mathrm{Err}^{\mathrm{UDISJ1}}_{N'}(\Pi) = \mathrm{Err}^{\mathrm{PDEP1}}_{N,2K,R}(\Pi').$$

Denote

$$\alpha' = 1 - \mathrm{Err}^{\mathrm{UDISJ1}}_{N'}(\Pi),\ \beta' = \mathrm{Err}^{\mathrm{UDISJ0}}_{N'}(\Pi),$$

and $\gamma' = \alpha' - \beta'$. We have

$$\mathrm{Adv}^{\mathrm{UDISJ}}_{N'}(\Pi) = \alpha' - \beta' = \gamma' =$$
$$1 - \mathrm{Err}^{\mathrm{PDEP1}}_{N,2K,R}(\Pi') - \mathrm{Err}^{\mathrm{PDEP0}}_{N,2K,R}(\Pi') = \mathrm{Adv}^{\mathrm{PDEP}}_{N,2K,R}(\Pi'),$$

where we assume that $\alpha' - \beta' \geq 0$ (otherwise, $\mathcal{A}, \mathcal{B}$ in Π simply negate the output of Π'). Hence, γ' is equal to the advantage of both the unique-disjointness and permutation-dependence protocols.

Let M_1, M_2 be the constants defined in Corollary 1 and define

$$M = 2/3 \cdot \min\{M_1, M_2\}.$$

Note that since $K \leq N/3$, then $M \leq 2/3 \cdot M_1 \leq M_1 \cdot (1 - K/N)$, or $M_1 \geq M/(1 - K/N)$. Furthermore, $N' = \lfloor N/K \rfloor \geq N/K - 1$. Therefore,

$$M_1 \cdot N' \geq \frac{M}{1 - \frac{K}{N}} \cdot \left(\frac{N}{K} - 1\right) = M \cdot N/K, \text{ and similarly } M_2 \cdot N' \geq M \cdot N/K.$$

To conclude the proof, we show that $\gamma' \leq C \cdot K/(M \cdot N)$.

If $\gamma' < \log N'/(M_1 \cdot N')$, then,

$$\gamma' < \log N'/(M_1 \cdot N') < \log N/(M_1 \cdot N') \leq C \cdot K/(M \cdot N).$$

Otherwise, $\gamma' \geq \log N'/(M_1 \cdot N')$. We apply Corollary 1, and since C bounds the communication cost of Π is the worst case, we conclude that $C \geq M_2 \cdot N' \cdot \gamma'$. This gives

$$\gamma' \leq C/(M_2 \cdot N') \leq C \cdot K/(M \cdot N),$$

as required.

■

5 The Multi-Pass Streaming Switching Lemma

For a parameter $P \geq 1$, we consider a P-pass streaming algorithm which can access an input stream of Q elements P times at the same order. The P-pass algorithm attempts to distinguish between a stream chosen from a random permutation or from a random function. In our model, the algorithm interacts with an oracle that samples from one of the distributions defined below.

For $0 < K \leq N$, let $\mathcal{D}_N^{K \times R \otimes P}$ be the distribution over $[N]^{K \cdot R \cdot P}$ that is defined by a sampling procedure which first draws $x \xleftarrow{\$} \mathcal{D}_N^{K \times R}$ and then outputs $\underbrace{x \| x \| \ldots \| x}_{P}$. In case $R = 1$, we simply write $\mathcal{D}_N^{K \otimes P}$.

Theorem 5 (multi-pass switching lemma). *There exists a constant $0 < M < 1$ such that any S-bounded randomized P-pass algorithm $\mathcal{A}$ with access to a stream containing $\log N \leq Q \leq N/3$ elements drawn from $[N]$ via either a random permutation or a random function has a distinguishing advantage bounded by*

$$\mathrm{Adv}^{\mathrm{STR}}_{\mathcal{D}_N^{Q \otimes P}, \mathcal{D}_N^{1 \times Q \otimes P}}(\mathcal{A}) \leq \mathrm{Opt}^{\mathrm{STR}}_{\mathcal{D}_N^{Q \otimes P}, \mathcal{D}_N^{1 \times Q \otimes P}}(S) \leq \frac{\lceil \log Q \rceil}{M} \cdot \frac{P \cdot Q \cdot S}{N} < \frac{\log N}{M} \cdot \frac{P \cdot Q \cdot S}{N}.$$

The proof of Theorem 5 is based on the lemma below, which is a generalization of Lemma 1.

Lemma 2. *There exists a constant $0 < M < 1$ such that for any $K \leq N/3$ and $S \cdot R \geq \log N$,*

$$\mathrm{Opt}^{\mathrm{STR}}_{\mathcal{D}_N^{2K \times R \otimes P}, \mathcal{D}_N^{K \times 2R \otimes P}}(S) \leq \frac{1}{M} \cdot \frac{P \cdot S \cdot R \cdot K}{N}.$$

We omit the proof of Theorem 5, as it is essentially identical to the one of Theorem 4.

The proof of Lemma 2 uses the following proposition which generalizes Proposition 1.

Proposition 4. *For any $K \leq N$,*

$$\mathrm{Opt}^{\mathrm{STR}}_{\mathcal{D}_N^{K \times R \otimes P}, \mathcal{D}_N^{K/2 \times 2R \otimes P}}(S) \leq \mathrm{Opt}^{\mathrm{PDEP}}_{N,K,R}(S \cdot R \cdot P).$$

Proof. The proof is via a reduction from the permutation-dependence problem to (multi-pass) streaming, which is similar to the one of Proposition 1. The only difference is that in order to simulate the P-pass streaming algorithm, its state is communicated (at most) $R \cdot P$ times, hence the communication cost of the permutation-dependence protocol is bounded by $S \cdot R \cdot P$. ∎

Proof (of Lemma 2). Let M be the constant implied by Proposition 2. Based on Proposition 4 and Proposition 2 we have

$$\mathrm{Opt}^{\mathrm{STR}}_{\mathcal{D}_N^{2K \times R \otimes P}, \mathcal{D}_N^{K \times 2R \otimes P}}(S) \leq \mathrm{Opt}^{\mathrm{PDEP}}_{N,2K,R}(S \cdot R \cdot P) \leq \frac{1}{M} \cdot \frac{P \cdot S \cdot R \cdot K}{N}.$$

∎

6 Conclusions and Future Work

In this paper we proved an upper bound on the streaming distinguishing advantage between a random permutation and a random function, which is tight up to poly-logarithmic factors. Our proof is based on a reduction from communication complexity to streaming, and is tailored to a common cryptographic setting where the goal is to distinguish between two pre-fixed distributions of streams. The cryptographic setting is different from the typical worst-case setting of streaming problems, where there is much more freedom in choosing the stream distributions in reductions from communication complexity. In the future, it would be interesting to apply our techniques to additional streaming problems that are relevant to cryptography.

Finally, our bounds in the (multi-pass) streaming switching lemma (theorems 4 and 5) depend on the constant M, where $M = 2/3 \cdot \min\{M_1, M_2\}$ (see the proof of Proposition 2).[7] The constants M_1, M_2 are defined in Theorem 3 and should be derived according to communication cost lower bounds on the disjointness problem. While M_1 only influences the lower range for which our bound is applicable, the value of M_2 is important if one needs to use our switching lemma in concrete security proofs. Communication cost lower bounds for disjointness obtained by information statistics [1] typically have small constant overhead, but additional technical effort is required in order to estimate M_2 in our setting where the stronger results of [4, 10] are required. We leave this to future work.

References

1. Z. Bar-Yossef, T. S. Jayram, R. Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *J. Comput. Syst. Sci.*, 68(4):702–732, 2004.
2. M. Bellare and P. Rogaway. The Security of Triple Encryption and a Framework for Code-Based Game-Playing Proofs. In S. Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 409–426. Springer, 2006.
3. K. Bhargavan and G. Leurent. On the Practical (In-)Security of 64-bit Block Ciphers: Collision Attacks on HTTP over TLS and OpenVPN. In E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, and S. Halevi, editors, *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 456–467. ACM, 2016.
4. M. Braverman and A. Moitra. An information complexity approach to extended formulations. In D. Boneh, T. Roughgarden, and J. Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 161–170. ACM, 2013.

[7] We note that in case N is a power of 2, it is possible to eliminate the factor of $2/3$ and set $M = \min\{M_1, M_2\}$.

5. R. P. Brent. An improved Monte Carlo factorization algorithm. *BIT Numerical Mathematics*, 20(2):176–184, 1980.
6. C. Cachin and U. M. Maurer. Unconditional security against memory-bounded adversaries. In B. S. K. Jr., editor, *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings*, volume 1294 of *Lecture Notes in Computer Science*, pages 292–306. Springer, 1997.
7. A. Chattopadhyay and T. Pitassi. The story of set disjointness. *SIGACT News*, 41(3):59–85, 2010.
8. W. Dai, V. T. Hoang, and S. Tessaro. Information-Theoretic Indistinguishability via the Chi-Squared Method. In J. Katz and H. Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III*, volume 10403 of *Lecture Notes in Computer Science*, pages 497–523. Springer, 2017.
9. O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *J. ACM*, 33(4):792–807, 1986.
10. M. Göös and T. Watson. Communication Complexity of Set-Disjointness for All Probabilities. *Theory of Computing*, 12(1):1–23, 2016.
11. C. Hall, D. A. Wagner, J. Kelsey, and B. Schneier. Building PRFs from PRPs. In H. Krawczyk, editor, *Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings*, volume 1462 of *Lecture Notes in Computer Science*, pages 370–389. Springer, 1998.
12. R. Impagliazzo and S. Rudich. Limits on the Provable Consequences of One-Way Permutations. In D. S. Johnson, editor, *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 44–61. ACM, 1989.
13. J. Jaeger and S. Tessaro. Tight Time-Memory Trade-offs for Symmetric Encryption. Cryptology ePrint Archive, Report 2019/258, 2019. Accepted to EUROCRYPT 2019.
14. B. Kalyanasundaram and G. Schnitger. The Probabilistic Communication Complexity of Set Intersection. *SIAM J. Discrete Math.*, 5(4):545–557, 1992.
15. D. E. Knuth. *The Art of Computer Programming, Volume II: Seminumerical Algorithms*. Addison-Wesley, 1969.
16. I. Kremer, N. Nisan, and D. Ron. On Randomized One-Round Communication Complexity. *Computational Complexity*, 8(1):21–49, 1999.
17. E. Kushilevitz and N. Nisan. *Communication complexity*. Cambridge University Press, 1997.
18. N. Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4):449–461, 1992.
19. J. M. Pollard. A monte carlo method for factorization. *BIT Numerical Mathematics*, 15(3):331–334, 1975.
20. A. A. Razborov. On the Distributional Complexity of Disjointness. *Theor. Comput. Sci.*, 106(2):385–390, 1992.