# Average-Case to (Shifted) Worst-Case Reduction for the Trace Reconstruction Problem

**Ittai Rubinstein**
Qedma Quantum Computing, Tel Aviv, Israel

## Abstract

In the *trace reconstruction problem*, one is given many outputs (called *traces*) of a noise channel applied to the same input message $\mathbf{x}$, and is asked to recover the input message. Common noise channels studied in the context of trace reconstruction include the *deletion channel* which deletes each bit w.p. $\delta$, the *insertion channel* which inserts a $G_j$ i.i.d. uniformly distributed bits before each bit of the input message (where $G_j$ is i.i.d. geometrically distributed with parameter $\sigma$) and the symmetry channel which flips each bit of the input message i.i.d. w.p. $\gamma$.

De et al. and Nazarov and Peres [12, 20] showed that any string $\mathbf{x}$ can be reconstructed from $\exp(O(n^{1/3}))$ traces. Holden et al. [13] adapted the techniques used to prove this upper bound, to construct an algorithm for average-case trace reconstruction from the insertion-deletion channel with a sample complexity of $\exp(O(\log^{1/3} n))$. However, it is not clear how to apply their techniques more generally and in particular for the recent worst-case upper bound of $\exp(\widetilde{O}(n^{1/5}))$ shown by Chase [7] for the deletion channel.

We prove a general reduction from the average-case to smaller instances of a problem similar to worst-case and extend Chase's upper-bound to this problem and to symmetry and insertion channels as well. Using this reduction and generalization of Chase's bound, we introduce an algorithm for the average-case trace reconstruction from the symmetry-insertion-deletion channel with a sample complexity of $\exp(\widetilde{O}(\log^{1/5} n))$.

**2012 ACM Subject Classification** Theory of computation → Sample complexity and generalization bounds

**Keywords and phrases** Trace Reconstruction, Synchronization Channels, Computational Learning Theory, Computational Biology

**Digital Object Identifier** 10.4230/LIPIcs.ICALP.2023.102

**Category** Track A: Algorithms, Complexity and Games

**Related Version** *Full Version*: https://arxiv.org/abs/2207.11489

**Acknowledgements** We would like to thank Zachary Chase, Roni Con and Aviad Rubinstein for their helpful comments on previous versions of this paper. We would also like to thank Nina Holden, Robin Pemantle, Yuval Peres and Alex Zhai for their help in understanding their paper.

## 1 Introduction

The *symmetry-insertion-deletion* (SID) channel with bit-flip probability $\gamma \in [0, 1/2)$, insertion probability $\sigma \in [0, 1)$ and deletion probability $\delta \in [0, 1)$, takes as input a binary string $\mathbf{x} \in \{0, 1\}^n$. For each $j$, the $j$th bit of $\mathbf{x}$ is flipped w.p. $\gamma$ (we will sometimes think of this portion of the channel as replacing the $j$th bit of $\mathbf{x}$ with a random bit w.p. $2\gamma$). Then $G_j$ random uniform and independent bits are inserted before the $j$th bit of $\mathbf{x}$, where the random variables $G_j \geq 0$ are i.i.d. geometrically distributed with parameter $\sigma$. Then, each bit of the message is deleted independently with probability $\delta$. The output string $\widetilde{\mathbf{x}}$ is called a *trace*[1].

[1] The trace reconstruction problem was originally defined with only the deletion channel [2] (i.e. with $\gamma$ and $\sigma$ fixed to 0). The more general SID channels were first considered in the "open questions" of [18] and were further researched by Andoni et al. [1] and by De et al. [12].

50th International Colloquium on Automata, Languages, and Programming (ICALP 2023).
Editors: Kousha Etessami, Uriel Feige, and Gabriele Puppis; Article No. 102; pp. 102:1–102:20
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The *trace reconstruction problem* asks the following question: how many traces are necessary to reconstruct an unknown string $\mathbf{x}$?

The main motivation for studying trace reconstruction comes from computational biology, where one often tries to align several DNA sequences to a common ancestor, and it has been extensively researched since the early 2000's [2].

Perhaps the most natural and well-researched version of the trace reconstruction problem, is the *worst-case*, where the input string $\mathbf{x}$ is adversarially chosen. Holenstein et al. [15] established an upper bound of $\exp(\widetilde{O}(n^{1/2}))$ on its sample complexity. This was improved by Nazarov and Peres [20], and De, O'Donnell and Servedio [12] who simultaneously proved upper and lower bounds of $\exp(O(n^{1/3}))$ on the sample complexity of "mean-based" trace reconstruction techniques. Recently, Chase [7] improved on these methods by proving that a "non-linear" method can be used to solve the worst-case deletion-channel trace reconstruction problem with a far lower sample complexity of $\exp(\widetilde{O}(n^{1/5}))$.

De et al. and Nazarov and Peres's results were highly influential and mean-based separators are used as a central component in the analysis of many other versions of the trace reconstruction problem [5, 8, 10, 13, 16, 21]. However, so far, Chase's techniques have not been extended beyond worst-case trace reconstruction from a deletion channel. In particular, we note the coded [5, 10] and the average-case [13, 21] trace reconstruction problems.

The *average-case* trace reconstruction problem was introduced by Batu et al. [2]. In this problem, the input string $\mathbf{x}$ is chosen uniformly at random from $\{0,1\}^n$, and the reconstruction only needs to succeed w.h.p. over the choice of $\mathbf{x}$. McGregor et al. [17] showed that if $H(n)$ traces are necessary for the worst-case trace reconstruction, then at least $H(\log n)$ are needed for the average-case (and under some conditions $H(\log n)\log n$).

Peres and Zhai [21] adapted mean-based separators to the average-case, constructing an efficient algorithm for the average-case deletion-channel trace reconstruction with $\exp(O(\log^{1/2} n))$ samples and low deletion probability ($\delta \leq 1/2$). This was further improved by Holden et al. [14] who reduced the sample complexity to $\exp(O(\log^{1/3} n))$ and generalized the algorithm to work for all insertion-deletion channels.

Motivated by the question of DNA storage, Cheraghchi et al. [10] introduced the *coded* trace reconstruction problem, where one is asked to construct a code $C \subset \{0,1\}^n$ s.t. any codeword $\mathbf{x} \in C$ can be reconstructed w.h.p. given as few independent traces $\widetilde{\mathbf{x}}$ as possible. Brakensiek et al. [5] proved that this problem is essentially equivalent to the average-case trace reconstruction problem.

## 1.1 Our Contributions

Let $n \in \mathbb{N}$ be arbitrarily large, and let $\gamma \in [0,1/2)$, $\sigma \in [0,1)$ and $\delta \in [0,1)$ be fixed bit-flip, insertion and deletion probabilities, and let $\mathcal{C}$ be the SID channel with these parameters. Let $C$ be a sufficiently large constant[2].

We introduce a new version of the trace reconstruction problem, called the *shifted* trace reconstruction problem (see Definition 1). In this problem, one is asked to reconstruct the first $n$ bits of a much longer string $\mathbf{x}$ from its traces. Moreover, the error channel is also allowed to "shift" the traces by some unknown distance $s \in \mathbb{N}$ (selected i.i.d. from a known and bounded distribution $S$ for each trace).

The shifted trace reconstruction problem often appears as a component in the analysis of other versions of the trace reconstruction problem [14, 8], but so far it has not been formally defined. Moreover, it could be of interest in its own right. Similar to the approximate trace reconstruction problem introduced by Davies et al. [11] and the average-case approximate

[2] $C$ may depend on $\gamma, \sigma, \delta$, but not on $n$.

trace reconstruction problem by Chase and Peres [8] which model the question of using a smaller number of traces to reconstruct some information about the input string $\mathbf{x}$, the shifted trace reconstruction problem asks a similar question, but with the goal of reconstructing the prefix of a long string.

▶ **Definition 1** (Shifted Trace Reconstruction Problem). *In a* shifted trace reconstruction problem *of size $n \in \mathbb{N}$, with* shift inaccuracy $\Delta_S(n)$*, one must reconstruct the $n+1$th bit of any string $\boldsymbol{x} \in \{0,1\}^{\mathbb{N}}$ of length at most $2^n$ given the value of its first $n$ bits $\boldsymbol{x}_{:n}$, and $H(n) = \exp(h(n))$ i.i.d. traces $\widetilde{\boldsymbol{x}}$ produced by the following process.*

*A random shift $s$ is applied to the input string $\overline{\boldsymbol{x}} \overset{\text{def}}{=} \boldsymbol{x}_{s:}$, where $s \leftarrow S$ is drawn from a known* shift distribution $S$*, with bounded support* $\mathrm{Supp}(S) \subseteq [a, a+\Delta_S]$ *for some $0 \leq a \leq n - \Delta_S$. Then the noise channel $\mathcal{C}$ is applied to the shifted string $\overline{\boldsymbol{x}}$ to obtain a trace $\widetilde{\boldsymbol{x}}$.*

The shifted trace reconstruction problem is clearly at least as hard as the worst-case trace reconstruction problem, but the differences between the two do not seem to affect the leading reconstruction techniques. In particular, we extend Chase's analysis to SID channels and to the shifted trace reconstruction problem, proving that $\exp(\widetilde{O}(n^{1/5}))$ samples suffice for the (shifted) worst-case trace reconstruction problem from an SID channel (Theorem 2).

▶ **Theorem 2.** *For any SID channel $\mathcal{C}$ as defined above, and for any constant $C > 0$, there exists an algorithm $A$ which solves the shifted trace reconstruction problem of size $n$ with shift inaccuracy $\Delta_S(n) = Ch(n)$ and sample complexity $H(n) = \exp(h(n)) = \exp(O(n^{1/5} \log^7 n))$.*

*Furthermore, when the deletion probability is sufficiently low ($\delta < 1/2$), the algorithm $A$ runs in time $\exp(O(n^{4/5} \log n))$ and if $q \geq 1/2$, then $A$ runs in time $\exp(O(n))$.*

▶ Remark 3. Note that while De et al.'s reconstruction algorithm has a time complexity polynomial in its sample complexity, Chase only proves an upper bound on the sample complexity. A naïve adaptation of Chase's upper bound to an algorithm would yield a time complexity of $\exp(\Theta(n))$.

Holden et al.'s average-case trace reconstruction algorithm works by partially aligning each trace and then using an oracle that solves a version of the shifted trace reconstruction problem to reconstruct each bit of the input message $\mathbf{x}$. However, much of their analysis is specific to their sample complexity of $\exp(\log^{1/3}(n))$.

We transform Holden et al.'s construction into a general reduction from an average-case trace reconstruction of length $n$ to linearly many instances of shifted trace reconstruction problems of length $O(\log(n))$ (Theorem 4). Moreover, our reduction applies to any SID channel.

▶ **Theorem 4** (Average to Shifted Reduction). *Let $A$ be an oracle that solves the shifted trace reconstruction problem with sample complexity $H(n) = \exp(h(n))$ (for $\log(n) \leq h(n) \leq \sqrt{n}$), shift inaccuracy $\Delta_S = Ch(n)$, and failure probability $< \exp(-n)$.*

*Then there exists an algorithm $A'$ that solves the average-case trace reconstruction problem with success probability $1 - o_n(1)$, sample complexity of $\exp(Ch(C \log n))$, and time complexity $t(n) = n^{1+o(1)}$, given $n$ calls to the oracle $A$.*

▶ Remark 5. Note that the assumption that $\log(n) \leq h(n) \leq \sqrt{n}$ is not very restrictive, since we show an upper bound of $h(n) \leq \widetilde{O}(n^{1/5})$ and the lower bound by Chase [6] implies that

$$h(n) \geq \frac{3}{2} \log n \ .$$

It is interesting to compare Theorem 4 to [17, Lemma 10]. Theorem 4 shows that if $H(n)$ traces suffice for shifted trace reconstruction then $\mathrm{poly}(H(\log n))$ traces suffice for average-case trace reconstruction. Compare this to McGregor et al. [17, Lemma 10] who prove that if $H(n)$ traces are required for worst-case trace reconstruction, then $H(\log n)$ traces are required for average-case trace reconstruction. This means that up to the differences between shifted and worst-case trace reconstruction, Theorem 4 is essentially tight.

We also note Brakensiek et al. [5], who proved reductions between the coded and the average-case trace reconstruction problems. When combined with Theorem 4 and [17, Lemma 10], a computational class of trace reconstruction problems begins to emerge (see Figure 1).

An important question to consider in future trace reconstruction research is whether other versions of the trace reconstruction problem can be reduced to one of these classes. For instance, consider the approximate average-case trace reconstruction problem. The best known approximate average-case trace reconstruction technique at the time of writing this paper is due to Chase and Peres [8], whose approach is based on performing calls to a "shifted" average-case trace reconstruction oracle, making it a good candidate for a more general reduction.

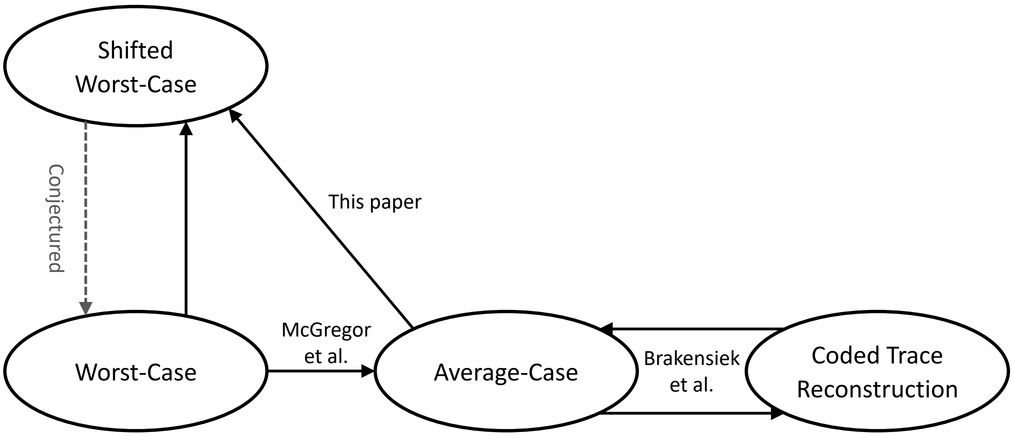

**Figure 1** A diagram of several known reductions between trace reconstruction problems. McGregor et al. [17] proved that any solution to the average-case trace reconstruction problem implies a solution to a smaller instance of the worst-case trace reconstruction problem. Brakensiek et al. [5] proved reductions from coded trace reconstruction to average-case trace reconstruction and vice versa. We introduce the shifted trace reconstruction and prove a reduction from the average-case to it. We also show that the current best-known solutions for worst-case trace reconstruction can be extended to shifted trace reconstruction and conjecture that the two are equivalent.

Finally, Theorems 2 and 4 give us an algorithm for the average-case trace reconstruction from SID channels with only $\exp(\widetilde{O}(\log^{1/5} n))$ traces.

▶ **Theorem 6** (Main Result). *For any SID channel $\mathcal{C}$ as defined above, if $\boldsymbol{x} \in \{0,1\}^n$ is a bit-string where the bits are chosen uniformly and independently at random, then we can reconstruct $\boldsymbol{x}$ with probability $1 - o_n(1)$ using $\exp(C \log^{1/5} n \log^7 \log n)$ traces. Moreover, when the deletion probability is sufficiently low ($\delta < 1/2$), this can be done in $n^{1+o(1)}$ time and otherwise, this can be done in polynomial time.*

## 1.2 An Overview of Previous Constructions

Many trace reconstruction techniques follow a similar high-level pattern [7, 12, 14, 20, 21]. First, a combinatorial analysis allows us to equate some property of the original message $\mathbf{x}$ to a polynomial whose coefficients depend on the traces $\widetilde{\mathbf{x}}$. This polynomial is then analysed on a small sub-arc of the complex disk $\mathbb{D}$ using Borwein and Erdélyi's seminal research on Littlewood polynomials [3] or an extension of it [7], yielding a statistical test on the traces which can be used to reconstruct some property of the original message $\mathbf{x}$.

Our analysis will also follow a similar pattern, and many of its steps will be based on combinations and extensions of components used to prove previous results, so we begin with a short overview of these techniques.

For any $\mathbf{w} \in \{0,1\}^{\mathbb{N}}$, let $I_{\mathbf{w}} : \{0,1\}^{\mathbb{N}} \to \mathbb{R}$ be the function that maps a string $\mathbf{x}$ to 1 if it begins with the prefix $\mathbf{w}$ and otherwise maps it to 0. For any function $f : \{0,1\}^{\mathbb{N}} \to \mathbb{R}$, we define its indicator polynomial on $\mathbf{x}$ to be the polynomial $p_{f,\mathbf{x}}(z) = \sum_j f(\mathbf{x}_{j:})z^j \in \mathbb{R}[z]$.

De et al. and Nazarov and Peres [12, 20] show that the polynomial $p_{\mathbf{x}} \overset{\text{def}}{=} p_{I_1,\mathbf{x}}$ whose $j$th coefficient is the $j$th bit of the original message $\mathbf{x}_j$ and the polynomial $p_{\widetilde{\mathbf{x}}} \overset{\text{def}}{=} \mathbb{E}\left[p_{I_1,\widetilde{\mathbf{x}}}\right]$ whose $j$th coefficient is the average over the $j$th bits of the traces $\mathbb{E}\left[\widetilde{\mathbf{x}}_j\right]$ are essentially equivalent up to a parameter change:

$$p_{\widetilde{\mathbf{x}}}(\phi^{-1}(z)) \approx (1-\delta)(1-2\gamma)p_{\mathbf{x}}(z) \tag{1}$$

where $\phi(z) = \frac{(1-\sigma)(\delta+(1-\delta)z)}{1-\sigma z}$ is a Möbius transformation related to the channel parameters[3]. De et al. and Nazarov and Peres then consider points of the form $z = \exp(i\alpha)$ for small $-n^{-1/3} < \alpha < n^{-1/3}$. $p_{\widetilde{\mathbf{x}}}(\phi^{-1}(z))$ can be approximated at such points from a bounded number of traces, because $\left|\phi^{-1}(z)\right| < 1 + O(n^{-2/3})$ and the linear transformation mapping the traces $\widetilde{\mathbf{x}}$ to

$$p_{\widetilde{\mathbf{x}}}(\phi^{-1}(z)) = \sum_{1 \le j \le n} \left(\phi^{-1}(z)\right)^j \mathbb{E}\left[\widetilde{\mathbf{x}}_j\right]$$

has bounded coefficients $\left|\phi^{-1}(z)\right|^j \le \exp(O(n^{1/3}))$.

Borwein and Erdélyi [3] showed that for any polynomial $p(z)$ with $\{0, \pm 1\}$ coefficients, there exists some $z$ in this sub-arc $\{\exp(i\alpha) \mid |\alpha| \le n^{-1/3}\}$ for which $|p(z)| \ge \exp(-O(n^{1/3}))$. In the context of trace reconstruction, we take $p(z)$ to be the difference $p_{\mathbf{x}}(z) - p_{\mathbf{y}}(z)$ where $\mathbf{x}$ and $\mathbf{y}$ are two input messages between which we want to differentiate.

This yields a method of differentiating between any two potential input strings $\mathbf{x}, \mathbf{y}$ with $\exp(O(n^{1/3}))$ traces.

Peres and Zhai [21] and Holden et al. [13] use a similar relationship between the original message and the traces, but in their construction the polynomials $p_{\mathbf{x}}$ and $p_{\widetilde{\mathbf{x}}}$ have a much higher degree because they want to reconstruct the first bits of a long string. They overcome this by extending the complex analysis to points of the form $z = \rho \exp(i\alpha)$ for a carefully chosen $\rho = 1 - o(1)$, effectively allowing them to truncate $p_{\mathbf{x}}$ and $p_{\widetilde{\mathbf{x}}}$ to a finite degree.

Holden et al. also use the fact that their input string is random to create partial alignments. The alignments are based on a Boolean test which checks whether or not a substring $\widetilde{\mathbf{w}}$ of a trace $\widetilde{\mathbf{x}}$ was the result of applying the channel to some substring $\mathbf{w}$ of the input message $\mathbf{x}$. This Boolean test is guaranteed to have a low false-positive rate and a non-negligible true-positive rate, when the input string $\mathbf{w}$ is "sufficiently random".

[3] Equation (1) is correct up to minor technical details. Lemma 8 can be used to derive an accurate version of this equation.

Holden et al. use this alignment procedure to reconstruct $\mathbf{x}$ one bit at a time. For each bit, they use this partial alignment to pin the traces to some nearby index and then use a mean-based separator to reconstruct it.

Chen et al. [9] and Narayanan and Ren [19] generalize equation (1) to relate multi-indices where some subsequence $\mathbf{w}$ appears in the input message $\mathbf{x}$ to multi-indices where the same subsequence appears in the traces, but their proof is limited to the deletion channel. As a results, Chase's analysis [7] which relies heavily on this generalized relationship, cannot be directly extended to insertion or symmetry channels.

Chase sets $\mathbf{w}$ to be an "a-periodic" string, in order to ensure that the set of indices where $\mathbf{w}$ appears as a consecutive substring in $\mathbf{x}$ is sparse. Therefore, the polynomial $p_{\mathbf{w},\mathbf{x}}(z) \overset{\text{def}}{=} p_{I_{\mathbf{w}},\mathbf{x}}(z)$ has sparse coefficients. Chase uses an extension of Borwein and Erdélyi's methods to prove much stronger bounds on polynomials with sparse coefficients on similar arcs of the unit disk. Balancing out the parameters yields Chase's $\exp(\widetilde{O}(n^{1/5}))$ bound on the worst-case sample complexity.

Much of our paper will be devoted to generalizing and combining the results of Holden et al. [14] and Chase [7]. For the sake of brevity, we will henceforth refer to these papers as the HPPZ and the Chase constructions respectively.

## 1.3 Sketch of our Proof

We extend these analyses in three key ways.

In the first and most difficult portion of the paper, we extend Chen et al. and Narayanan and Ren's [9, 19] generalization of equation (1) to SID channels. This is non-trivial, as the common method of dealing with insertions and bit-flips is to take a statistic where the unbiased insertions average out to having no effect on the output (this is usually done by looking at the difference between the traces $\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}}$ of two potential input strings $\mathbf{x}, \mathbf{y}$). It is not clear how to perform a similar analysis on a multi-bit property, such as an indicator of some magic string $\mathbf{w}$ which is highly non-linear in its input.

Our main observation in dealing with this problem is that the function $\chi_{(-1,\ldots,-1)}(\mathbf{x}) \overset{\text{def}}{=} (-1)^{\mathbf{x}_1} \cdots (-1)^{\mathbf{x}_k}$, which we call a "full" character, has the property that if any of its input bits $\mathbf{x}_1, \ldots, \mathbf{x}_k$ is an inserted bit or was replaced with a random bit by the symmetry channel, then its output is unbiased and does not affect the average over traces. This allows us to prove a similar relationship between $p_{\chi_{(-1,\ldots,-1)},\mathbf{x}}$ and $p_{f,\tilde{\mathbf{x}}}$.

Next, we extend this analysis to all characters $f(\mathbf{x}) = \chi_\omega(\mathbf{x}) \overset{\text{def}}{=} \prod_j \omega_j^{\mathbf{x}_j}$ for $\omega \in \{\pm 1\}^k$. This extension is more complex and requires several difficult technical lemmas, but it allows us to reconstruct $p_{f,\mathbf{x}}(z)$ for (almost) any function $f : \{0,1\}^k \to \mathbb{R}$ from the traces. We do this by applying the Fourier transformation on Boolean functions to $f$, allowing us to write $f(\mathbf{x}) = \sum_\omega \chi_\omega(\mathbf{x})\widehat{f}(\omega)$ as a linear combination of characters, and by extension

$$p_{f,\mathbf{x}}(z) = p_{\sum_\omega \chi_\omega(\cdot)\widehat{f}(\omega),\mathbf{x}}(z) = \sum_\omega \widehat{f}(\omega) p_{\chi_\omega,\mathbf{x}}(z).$$

In the second portion of our analysis, we extend the Borwein and Erdélyi-type bounds proven by Chase [7] to deal with sparse polynomials when evaluated at points within the unit disk. This step is necessary for our extension of Chase's bounds to the shifted trace reconstruction problem.

In the third and final portion of the paper, we generalize Holden et al.'s [13] construction into a reduction from an average-case trace reconstruction problem of size $n$ to linearly many trace reconstruction problems of size $\Theta(\log(n))$. Moreover, we extend Holden et al.'s proofs originally shown for the insertion-deletion channel to SID channels as well.

## 1.4 Organization of the Paper

Sections 2 and 3 contain the heart of our analysis, where we convert the shifted trace reconstruction problem into a complex analysis one (2) and use complex analysis techniques to solve it (3). We adapt Holden et al.'s techniques to prove a general reduction in Section 4, proving Theorems 4 and 6. Section 5 is reserved for a discussion of our results.

# 2 Conversion to Complex Analysis

Let $\mathbf{x} \in \{0,1\}^{\mathbb{N}}$ be some input string and let $\widetilde{\mathbf{x}}$ denote its trace from some shifted trace reconstruction problem. The first step of our analysis will be to relate a property of $\mathbf{x}$ to the expectation of some function applied to its traces $\widetilde{\mathbf{x}}$. By bounding this function of the traces in absolute value, we prove that this function can be approximated from a bounded number of traces. Then, in Section 3, we will show that approximating this property of the input string $\mathbf{x}$ allows us to reconstruct $\mathbf{x}$ one bit at a time.

This approach to trace reconstruction is common in recent literature. De et al., Holden et al. and Nazarov and Peres [12, 14, 20] show how "single bit statistics" of the input string $\mathbf{x}$ can be related to its traces through SID channels and shifts. Chase [7], building off of the works of Chen et al. [9] and Narayanan and Ren [19], extended this relationship to multi-bit statistics, in order to prove a stronger bound on the sample complexity of trace reconstruction.

However, Chase's analysis is limited to deletion channels and the main known tools for dealing with insertions and bit-flips are inherently limited to single-bit statistics. Our goal in this section will be to combine these approaches, allowing us to estimate multi-bit properties of the input string $\mathbf{x}$ from traces through an SID channel.

Let $1 \le \ell \le 2n^{1/5} + 1$ be some integer. For any function $f : \{0,1\}^\ell \to \mathbb{D}$ from the hypercube to the unit disk $\mathbb{D}$ (for our use-case, we will want $f = I_{\mathbf{w}}$ to be the indicator of some marker $\mathbf{w} \in \{0,1\}^{\mathbb{N}}$), we define $q_{f,\mathbf{x}}(z_0, \ldots, z_\ell)$ to be

$$q_{f,\mathbf{x}}(z_0,\ldots,z_\ell) \overset{\text{def}}{=} \sum_{k_0<k_1<\cdots<k_\ell} (-1)^{\mathbf{x}_0} z_0^{k_0} f(\mathbf{x}_{k_1},\ldots,\mathbf{x}_{k_\ell}) \prod_{1\le j\le \ell} z_j^{k_j-k_{j-1}-1}.$$

In essence, $q_{f,\mathbf{x}}(z_0, \ldots, z_\ell)$ is a multivariate polynomial whose coefficients encode the value of $f$ when applied to subsequences of the input string $\mathbf{x}$. Our goal will be to show that the value of this polynomial can be estimated at certain points from a bounded number of traces (see Theorem 7).

▶ **Theorem 7.** *Let $z_0$ be a point on the arc $\left\{(1 - n^{-4/5}\log^6 n)\exp(i\alpha) \mid \alpha \in [-n^{-2/5}, n^{-2/5}]\right\}$. If $\delta < 1/2$, let $z_1, \ldots, z_\ell = 0$. Otherwise, let $z_1 = \cdots = z_\ell$ be any point in the segment $[1 - c_1, 1 - c_2]$ for sufficiently small constants $c_1 > c_2 > 0$.*

*Given $H(n) = \exp(h(n))$ traces with shift inaccuracy $\eta = O(h(n))$, we can estimate $q_{f,x}$ at the point $(z_0, \ldots, z_\ell)$ to within an additive error of order $\pm\exp(-\Omega(h(n)))$ and with success probability $1 - \exp(-\omega(n))$, where $h(n) = n^{1/5}\log^7 n$.*

We separate the proof of Theorem 7 into three parts. In the first part of the proof (Lemma 8), we will show that the statement holds for the function $f(\mathbf{x}_1, \ldots, \mathbf{x}_\ell) = \prod_{1\le j\le \ell}(-1)^{\mathbf{x}_j}$. We will call this function a "full character".

We will then extend this proof to any character $f(\mathbf{x}_1, \ldots, \mathbf{x}_\ell) = \chi_\omega(\mathbf{x}_1, \ldots, \mathbf{x}_\ell) = \prod_{1\le j\le \ell}\omega_j^{\mathbf{x}_j}$ (with $\omega \in \{\pm 1\}^\ell$) of the Fourier transformation on Boolean functions. Finally, to complete the proof, we will use the fact that any function $f : \{0,1\}^\ell \to \mathbb{D}$ can be written as a linear combination of characters $f = \sum_\omega \hat{f}(\omega)\chi_\omega$ (where $\hat{f}$ is the Fourier transformation of $f$).

▶ **Lemma 8.** *Let $S$ be a shift distribution with bounded support (*$\operatorname{Supp}(S) \subseteq \{0,1,\dots,d\}$*). Let $\boldsymbol{x} \in \{0,1\}^{\mathbb{N}}$ be an input string, and let $\widetilde{\boldsymbol{x}}$ be the trace of $\boldsymbol{x}$, sampled by applying the SID channel with deletion probability $\delta$, insertion probability $\sigma$ and bit-flip rate $\gamma$ applied to the randomly shifted string $\boldsymbol{x}_{s:}$ (where $s \leftarrow S$).*

*Define $\phi_1(z) \overset{\text{def}}{=} (1-\delta)z+\delta$, $\phi_2(z) \overset{\text{def}}{=} \frac{(1-\sigma)z}{1-\sigma z}$, $\phi \overset{\text{def}}{=} \phi_2 \circ \phi_1$. For all $j$, we set $\zeta_j = \phi^{-1}(z_j)$.*

*Define $P(z) \overset{\text{def}}{=} \sum_{s=0}^{d} \Pr[S=s] z^s$. Then:*

$$\begin{aligned} P(z_0^{-1}) \sum_{k_0<k_1<\cdots<k_\ell} (-1)^{\boldsymbol{x}_0} z_0^{k_0} \prod_{1\le j\le \ell} z_j^{k_j-k_{j-1}-1} (-1)^{\boldsymbol{x}_j-\boldsymbol{x}_{j-1}-1} = \\ = \left( \prod_{0\le j\le \ell} \frac{\phi_1(\zeta_j)}{(1-\delta)(1-2\gamma)\zeta_j} \right) \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}} \left[ \sum_{r_0<\cdots<r_\ell} \zeta_0^{r_0} (-1)^{\tilde{\boldsymbol{x}}_{r_0}} \left( \prod_{j=1}^{\ell} (-1)^{\tilde{\boldsymbol{x}}_{r_j}} \zeta_j^{r_j-r_{j-1}-1} \right) \right] \end{aligned} \tag{2}$$

Lemma 8 moves us closer to the goal of proving Theorem 7, because the left-hand-side of equation (2) is essentially equivalent to $q_{f,\mathbf{x}}(z_0,\dots,z_\ell)$ for the function $f(\mathbf{x}_1,\dots,\mathbf{x}_\ell) = \prod_{1\le j\le \ell}(-1)^{\mathbf{x}_j}$, while the right-hand-side depends only on the traces.

## 2.1 Proof of Lemma 8

We begin by proving Lemma 8 for the simpler case, where the shift $s$ and the bit-flip probability are both fixed to 0.

Let $D_k$ denote the event that the $k$th bit of the input string $\mathbf{x}$ was *not* deleted by the channel. Conditioned on $D_k$, let $R_k$ be the distribution of the index $r_k$ within the trace $\widetilde{\mathbf{x}}$ to which this bit was mapped. For any distribution $V$, let $G_V(\zeta) \overset{\text{def}}{=} \sum_v \Pr[V=v]\zeta^v$ denote its generating function.

Consider the generating function $G_{R_k}(\zeta)$ of $R_k$. Each of the first $k$ bits of the input message $\mathbf{x}$ was expanded to an i.i.d. geometrically distributed number of bits, and then each of those was either retained or deleted, resulting in a Bernoulli distribution of bits (except for the last bit which was not deleted, because we conditioned on $D_k$). Using common identities on products and compositions of generating functions, we derive equation (3).

$$G_{R_k}(\zeta) = \left(G_{\mathrm{Geom}(\sigma)}\left(G_{\mathrm{Bern}(\delta)}(\zeta)\right)\right)^{k-1} G_{\mathrm{Geom}(\sigma)-1}\left(G_{\mathrm{Bern}(\delta)}(\zeta)\right)\zeta = \frac{\zeta}{\phi_1(\zeta)}\phi(\zeta)^k \tag{3}$$

Denote by $I_r$ the event that the $r$th bit of the trace was an insertion. Conditioned on $I_r$, the $r$th bit of $\widetilde{\mathbf{x}}$ is a Bernoulli$\left(\frac{1}{2}\right)$ random variable independent of the rest of the problem. Consider the expectation of $f(\widetilde{\mathbf{x}}_{r_0},\dots,\widetilde{\mathbf{x}}_{r_\ell}) = \prod_{1\le j\le l}(-1)^{\tilde{\mathbf{x}}_{r_j}}$ over the traces. Due to our choice of $f$, if even one of its inputs is an insertion, then its expectation is $\mathbb{E}_{\tilde{\mathbf{x}}}\left[\prod_{0\le j\le l}(-1)^{\tilde{\mathbf{x}}_{r_j}} \middle| I_{r_j}\right] = 0$.

The event that the $r$th bit of the trace $\widetilde{\mathbf{x}}_r$ was not due to an insertion is exactly equal to the event that some bit $\mathbf{x}_k$ in the input message was not deleted ($D_k$) and that it was transmitted as the $r$th bit of the trace ($R_k = r$). Therefore, the expectation of $f$ on the multi-index $r_0,\dots,r_\ell$ of the trace is equal to

$$\begin{aligned} \mathop{\mathbb{E}}_{\tilde{\mathbf{x}}}\left[\prod_{0\le j\le l}(-1)^{\tilde{\mathbf{x}}_{r_j}}\right] &= \sum_{k_0<\cdots<k_\ell}(-1)^{\mathbf{x}_{k_j}}\Pr\left[\bigwedge_{0\le j\le \ell}\left(D_{k_j}\wedge\left(R_{k_j}=r_j\right)\right)\right] \\ &= (1-\delta)^\ell \sum_{k_0<\cdots<k_\ell}(-1)^{\mathbf{x}_{k_j}}\Pr\left[\bigwedge_{0\le j\le \ell}\left(R_{k_j}=r_j\right)\middle|\bigwedge_{0\le j\le \ell}D_{k_j}\right] \end{aligned} \tag{4}$$

Finally, note that given $D_{k_j}$ and the value of $r_j = R_{k_j}$, the effect of the channel on the next bits is independent of $r_j$. Therefore, conditioned on $D_k$ and $D_{k+1}$, we have

$$\Pr_{\tilde{\mathbf{x}}}\left[R_{k_j} = r_j \,\middle|R_{k_{j-1}} = r_{j-1}\right] = \Pr_{\tilde{\mathbf{x}}}\left[R_{k_j-k_{j-1}-1} = r_j - r_{j-1} - 1\right]. \tag{5}$$

Combining equations (3), (4) and (5), we see that

$$\begin{aligned}
&(1-\delta)^{-\ell}\mathop{\mathbb{E}}_{\tilde{\mathbf{x}}}\left[\sum_{r_0<\cdots<r_\ell}(-1)^{\tilde{\mathbf{x}}_{r_0}}\zeta_0^{r_0}\prod_{1\le j\le l}(-1)^{\tilde{\mathbf{x}}_{r_j}}\zeta_j^{r_j-r_{j-1}-1}\right]= \\
&=\sum_{r_0<\cdots<r_\ell}\sum_{k_0<\cdots<k_\ell}\Pr\left[\bigwedge_{0\le j\le\ell}(R_{k_j}=r_j)\,\middle|\,\bigwedge_{0\le j\le\ell}D_{k_j}\right] \\
&\qquad\qquad(-1)^{\mathbf{x}_{k_0}}\zeta_0^{r_0}\prod_{1\le j\le l}\zeta_j^{r_j-r_{j-1}-1}(-1)^{\mathbf{x}_{k_j}}= \\
&=\sum_{k_0<\cdots<k_\ell}\frac{\zeta_0}{\phi_1(\zeta_0)}\phi(\zeta_0)^{k_0}(-1)^{\mathbf{x}_{k_0}}\prod_{1\le j\le l}\frac{\zeta_j}{\phi_1(\zeta_j)}\phi(\zeta_j)^{k_j-k_{j-1}-1}(-1)^{\mathbf{x}_{k_j}}
\end{aligned} \tag{6}$$

Some minor manipulations to equation (6), yields equation (2) for the case when $s$ and $\gamma$ are fixed to 0. Finally, we extend the proof to shifts and bit-flips. Let $\overline{\mathbf{x}}$ denote the output of the shift and symmetry portions of the channel. It is easy to show that

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\overline{\mathbf{x}}}\left[\sum_{k_0<\cdots<k_\ell}z_0^{k_0}(-1)^{\overline{\mathbf{x}}_{k_0}}\prod_{1\le j\le l}z_j^{k_j-k_{j-1}-1}(-1)^{\overline{\mathbf{x}}_{k_j}}\right]= \\
&\quad=(1-2\gamma)^{\ell}\mathop{\mathbb{E}}_{s\leftarrow S}\left[\sum_{k_0<\cdots<k_\ell}z_0^{k_0}(-1)^{\mathbf{x}_{k_0+s}}\prod_{1\le j\le l}z_j^{k_j-k_{j-1}-1}(-1)^{\mathbf{x}_{k_j+s}}\right]= \\
&\quad=(1-2\gamma)^{\ell}P\left(\frac{1}{z_0}\right)\sum_{k_0<\cdots<k_\ell}z_0^{k_0}(-1)^{\mathbf{x}_{k_0}}\prod_{1\le j\le l}z_j^{k_j-k_{j-1}-1}(-1)^{\mathbf{x}_{k_j}}
\end{aligned} \tag{7}$$

Combining equations (6) and (7) yields Lemma 8.

## 2.2 Sketch of the Proof of Theorem 7

Due to space limitations, we reserve the rest of the proof of Theorem 7 to the full version of the paper which can be found on arxiv [23], where we show that Lemma 8 implies Theorem 7. The rest of this section is devoted to giving the high-level idea of this proof.

The first step of the proof is an analysis of the Möbius transformations in Lemma 8. In particular, we show that for the points $z_0, \ldots, z_\ell$ chosen as in Theorem 7, the absolute values of $\zeta_0, \ldots, \zeta_\ell$ are bounded below 1.

This allows us to truncate the RHS of equation (2) to only its low degree terms with a negligible effect on the output. This truncation enables us to evaluate this polynomial at the required points, proving Theorem 7 for the full character $f(\mathbf{x}_1, \ldots, \mathbf{x}_\ell) = \prod_j (-1)^{\mathbf{x}_j}$.

In fact, this allows us to estimate $q_{f,\mathbf{x}}(z_0, \ldots, z_\ell)$ for any choice of $z_1, \ldots, z_\ell$ sufficiently close to those defined Theorem 7 when $f$ is the full character. We use this fact to prove Theorem 7 for general characters. In essence, we show that $q_{\chi_\omega,\mathbf{x}}(z_0, \ldots, z_\ell)$ can be written as a high-order derivative of $q_{f',\mathbf{x}}$ for a full character $f'$ on fewer bits $\ell' < \ell$ and that this derivative can be approximated from a limited number of samples using Lemma 9.

▶ **Lemma 9.** *Let $c, \delta > 0$ be some real parameters and let $P$ be an oracle that computes for a given point $z_1, \ldots, z_l \in [-c, c]^l$ the value of some polynomial $p$ of degree at most $n$ at the given point, up to some additive error $\delta > 0$. Let $\boldsymbol{j} = (j_1, \ldots, j_l)$ be some vector of integers (all smaller than $n$), define $m_{\boldsymbol{j}} = z_1^{j_1} \cdots z_l^{j_l}$ be the $\boldsymbol{j}$th monomial and $j_{\text{tot}} = \sum_i j_i$.*

*Given* $\text{poly}(n, 1/c)^{O(l+j_{\text{tot}})}$ *queries to $P$, we can compute the coefficient of $m_{\boldsymbol{j}}$ to within an additive error of* $\text{poly}(n, 1/c)^{O(l+j_{\text{tot}})}\delta$ *in time* $\text{poly}(n, 1/c)^{O(l+j_{\text{tot}})}$.

Finally, we use the fact that $q_{f,\mathbf{x}}$ is linear in our choice of $f$ (i.e. for any $f_1, f_2$, $q_{f_1+f_2,\mathbf{x}} = q_{f_1,\mathbf{x}} + q_{f_2,\mathbf{x}}$), and the fact that any function $f$ can be written as a linear combination of character functions $f = \sum_{\omega \in \{\pm 1\}^\ell} \hat{f}(\omega)\chi_\omega$ via a Fourier transformation. Combining these observations, we see that

$$q_{f,\mathbf{x}} = \sum_{\omega \in \{\pm 1\}^\ell} \hat{f}(\omega) q_{\chi_\omega,\mathbf{x}}.$$

The RHS of this equation can be estimated from the traces (one element at a time) and the LHS was our original goal, thus proving Theorem 7.

## 3 Proof of Theorem 2

In Section 2, we showed that for any function $f$ from $\{0,1\}^\ell$ to the unit disk $\mathbb{D}$, we can map it into a polynomial related to the input message which can be approximated to a high degree of accuracy from the traces. In this section, we will construct a function $f$ for which our approximation of $q_{f,\mathbf{x}}$ as promised by Theorem 7 will suffice to reconstruct the $n+1$th bit of the input string $\mathbf{x}$, proving Theorem 2.

A central component of our analysis will be Theorem 10, which is a slight generalisation of [7, Theorem 5]. In this theorem we show that members of a certain class of polynomials have some non-negligible values on a sufficiently small sub-arc of the unit disk $\mathbb{D}$.

The polynomial $p(z)$ in Theorem 10 should be thought of as the difference between two polynomials $q_{f,\mathbf{x}}(z, 0, \ldots, 0) - q_{f,\mathbf{y}}(z, 0, \ldots, 0)$ for two hypotheses $\mathbf{x}$ and $\mathbf{y}$ for the input string. By proving that these polynomials differ at a point where they can be estimated from the traces, we show that this estimation can be used to differentiate between the hypotheses.

▶ **Theorem 10** (Extension of [7, Theorem 5])**.** *Let $\mathcal{P}_n^\mu$ denote the set of polynomials of the form $p(z) = \xi - \eta z^d + \sum_{n^\mu \le j \le n} a_j z^j$ where $\eta \in \{0,1\}$, $\xi \in \partial\mathbb{D}$ and $|a_j| \le 2$.*

*For any $\mu \in (0,1)$, there exists some constant $C > 0$, such that for all sufficiently large $n$, any $p \in \mathcal{P}_n^\mu$, it holds that for every $\rho \in [0,1]$:*

$$\max_{|\alpha| \le n^{-2\mu}} \left| p(\rho e^{i\alpha}) \right| \ge \exp\left(-C n^\mu \log^5 n\right)$$

Our proof of Theorem 10 is similar to Chase's proof of [7, Theorem 5], and due to space limitations, we reserve it for the full version of this paper [23]. Throughout the rest of this section, we will prove that Theorem 2 follows from Theorem 10.

In Section 3.1, we will extend Theorem 10 to a wider class of polynomials, proving that the estimation method described in Theorem 7 can be used to distinguish between the traces of any two potential input strings $\mathbf{x}, \mathbf{y}$. In Section 3.2, we will show how this distinguishing oracle can be used to reconstruct a string $\mathbf{x}$ from the shifted trace reconstruction problem, proving Theorem 2.

## 3.1 Corollaries of Theorem 10

In this section, we will extend Theorem 10 to prove that for any strings $\mathbf{x}, \mathbf{y}$ which agree on their first $n$ bits, the estimation oracle described in Theorem 7 can be used to distinguish between their traces. This proof will follow from two main components.

First, we will show that for any two such strings $\mathbf{x}, \mathbf{y}$, there exists some choice of indicator function $f = I_{\mathbf{w}}$, such that for $p_{f,\mathbf{x}}(z) = q_{f,\mathbf{x}}(z, 0, \ldots, 0)$, the polynomial $p_{\text{diff}}(z) = p_{f,\mathbf{x}}(z) - p_{f,\mathbf{y}}(z)$ (almost) fits the requirements of Theorem 10. Therefore, if $\delta < 1/2$, then there exists some point $(z, 0, \ldots, 0)$ such that we can estimate the evaluation of $q_{f,\mathbf{x}}$ from the traces and that $q_{f,\mathbf{x}}$ and $q_{f,\mathbf{y}}$ differ significantly at this same point. This yields a method of distinguishing between their traces (see Corollary 11).

Then, in Corollary 12, we will extend this analysis to higher deletion probabilities, by showing that a similar distinguishing method can also be used at points of the form $(z, 1-c, \ldots, 1-c)$. For the rest of this section, let $\mu = 1/5$, $\rho = 1 - n^{-4/5} \log^6 n$, $\ell = 2n^{1/5} + 1$, and $\mathcal{A} = \{\rho e^{i\alpha} \mid |\alpha| \leq n^{2/5}\}$.

The following is a corollary of Theorem 10:

▶ **Corollary 11** (Adaptation of Proposition 6.3 from [7]). *Let $\boldsymbol{x}, \boldsymbol{y} \in \{0,1\}^{\mathbb{N}}$ be binary strings that agree on their first $n$ bits ($\boldsymbol{x}_{:n} = \boldsymbol{y}_{:n}$) and disagree on their $(n+1)$th bit ($\boldsymbol{x}_{n+1} \neq \boldsymbol{y}_{n+1}$). Then there exist some $\boldsymbol{w} \in \{0,1\}^{\ell}$ and $z_0 \in \mathcal{A}$ such that*

$$|q_{I_{\boldsymbol{w}},\boldsymbol{x}}(z_0, 0, \ldots, 0) - q_{I_{\boldsymbol{w}},\boldsymbol{x}}(z_0, 0, \ldots, 0)| \geq \exp\left(-n^{1/5} \log^6 n\right) \exp\left(-Cn^{1/5} \log^5 n\right)$$

**Proof of Corollary 11.** Let $\mathbf{x}$ and $\mathbf{y}$ be two hypotheses for the input string to a shifted trace reconstruction problem (that agree on their first $n$ bits and not on their $n+1$th bit).

Let $\mathbf{w}' = \mathbf{x}(n - \ell + 1 : n)$. Lemmas 1 and 2 of [22] imply that at least one of $\mathbf{w}'0$ or $\mathbf{w}'1$ has no period of length $\leq n^{1/5}$ and that for this choice of $\mathbf{w} \in \{\mathbf{w}'0, \mathbf{w}'1\}$, the indices $k$ for which $\mathbf{x}_{k:k+\ell} = \mathbf{w}$ are $n^{1/5}$ separated.

Consider the polynomial

$$\begin{aligned} p_{\mathbf{w}}(z) &\overset{\text{def}}{=} z^{\ell-n-1} \left[q_{I_{\mathbf{w}},\mathbf{x}}(z, 0, \ldots, 0) - q_{I_{\mathbf{w}},\mathbf{y}}(z, 0, \ldots, 0)\right] \\ &= \sum_k \left[(-1)^{\mathbf{x}_k} 1_{\mathbf{x}(k+1:k+\ell)=\mathbf{w}} - (-1)^{\mathbf{y}_k} 1_{\mathbf{y}_{k+1:k+\ell}=\mathbf{w}}\right] z^{k+\ell-n-1} \end{aligned}$$

Because $\mathbf{x}$ and $\mathbf{y}$ agree on their first $n$ bits, $p_{\mathbf{w}}(z)$ has no negative powers. By our definition of $\mathbf{w}$ to be either $\mathbf{x}_{n-\ell+1:n+1}$ or $\mathbf{y}_{n-\ell+1:n+1}$, the 0th power of $p_{\mathbf{w}}(z)$ is $\pm 1$. Moreover, all of $p_{\mathbf{w}}(z)$'s coefficients are bounded by 2 in absolute value and its non-zero powers maintain the sparsity condition of Theorem 10.

The only problem with applying Theorem 10 to $p_{\mathbf{w}}(z)$ is that its degree is not bounded by $n$. We overcome this issue by defining $\widetilde{p_{\mathbf{w}}}(z)$ to be the truncation of $p_{\mathbf{w}}(z)$ to its $n$th power. Applying Theorem 10 to the polynomial $\widetilde{p_{\mathbf{w}}}$, we see that there exists a point $z_0 \in \mathcal{A}$ for which $|\widetilde{p_{\mathbf{w}}}(z_0)| \geq \exp\left(-C_1 n^{\mu} \log^5 n\right)$

Because we want to evaluate $\widetilde{p_{\mathbf{w}}}(z)$ at points $z$ with absolute value $|z| = \rho$ strictly below 1, we can also bound the effect of this truncation by

$$|\widetilde{p_{\mathbf{w}}}(z) - p_{\mathbf{w}}(z)| \leq \frac{\rho^n}{1-\rho} = \text{poly}(n) \exp(-n^{\mu} \log^6 n) = o\left(|\widetilde{p_{\mathbf{w}}}|\right)$$

From here we apply the triangle inequality to show that $|p_{\mathbf{w}}(z)| \geq \exp\left(-C_2 n^{\mu} \log^5 n\right)$.

Finally, note that $|q_{I_{\mathbf{w}},\mathbf{x}}(z, 0, \ldots, 0) - q_{I_{\mathbf{w}},\mathbf{y}}(z, 0, \ldots, 0)| = |p_{\mathbf{w}}(z)||z|^{n-\ell+1}$, completing the proof of Corollary 11. ◀

Corollary 11 allowed us to use Theorem 7 to distinguish between the traces of any two string $\mathbf{x}$ and $\mathbf{y}$ when the deletion probability of the channel is low ($\delta < 1/2$).

However, this proof relied on our ability to estimate the value of $q_{I_\mathbf{w},\mathbf{x}}$ at points where $z_1 = \cdots = z_\ell = 0$, and when the deletion probability is high ($\delta \geq 1/2$), Theorem 7 only allows us to evaluate $q_{I_\mathbf{w},\mathbf{x}}$ at points of the form $z_1 = \cdots = z_\ell \in [1 - c_1, 1 - c_2]$. In order to distinguish between the traces of $\mathbf{x}$ and $\mathbf{y}$ from high deletion probability channels, we extend Theorem 7 to multivariate polynomials sampled at such points. We do this in Corollary 12.

▶ **Corollary 12** (Adaptation of Corollary 6.1 from [7]). *Let $c_1 > c_2 > 0$ be sufficiently small positive constants, and let $\boldsymbol{x}, \boldsymbol{y} \in \{0,1\}^{\mathbb{N}}$ be as in Corollary 11. There exist some $\boldsymbol{w} \in \{0,1\}^l$, $z_0 \in \mathcal{A}$ and $z_1 = \cdots = z_\ell \in [1 - c_1, 1 - c_2]$, such that*

$$|q_{I_{\boldsymbol{w}},\boldsymbol{x}}(z_0, z_1, \ldots, z_1) - q_{I_{\boldsymbol{w}},\boldsymbol{x}}(z_0, z_1, \ldots, z_1)| \geq \exp\left(-n^{1/5}\log^6 n\right)\exp\left(-Cn^{1/5}\log^5 n\right)$$

**Proof of Corollary 12.** Fix $\mathbf{w}$ and $z_0$ to be the same as in the proof of Corollary 11. We define $Q$ to be the following polynomial in $z_1$, for $z_1 \in [0, 1 - c_2]$.

$$Q(z_1) \overset{\text{def}}{=} (1-\rho)\binom{n}{\ell}^{-1}\left[q_{I_\mathbf{w},\mathbf{x}}(z_0, z_1, \ldots, z_1) - q_{I_\mathbf{w},\mathbf{y}}(z_0, z_1, \ldots, z_1)\right]$$

Consider the coefficient of the $j$th power of $z_1$ in $Q$. If $j \leq n$, then this coefficient is bounded by 1 in absolute value. This is because our summation over the powers of $z_0$ can contribute a factor of at most $1/(1-\rho)$, and the number of terms in $q_{I_\mathbf{w},\mathbf{x}}$ with total degree $j$ is at most $\binom{n}{\ell}$.

If $j > n$, then the number of monomials of $q_{I_\mathbf{w},\mathbf{x}}$ with total degree $j$ is at most $\exp(O(\ell\log(j)))$, but the value of the monomial $z_1^j$ is at most $(1-c_2)^j = \exp(-\Omega(j))$.

Therefore, truncating these higher powers of $Q$ would have a negligible effect on its value. Let $\widetilde{Q}(z_1)$ be the truncation of $Q$ to monomials of degree $\leq n$. $\widetilde{Q}$ is a univariate polynomial in $z_1$, with coefficients bounded from above by 1, and for any $z_1 \in [0, 1 - c_2]$, we have

$$\left|Q(z_1) - \widetilde{Q}(z_1)\right| \leq \exp(-\Omega(n)) \tag{8}$$

In Corollary 11, we showed that $|Q(0)|$ is bounded from below, and this lower bound can be naturally extended to $\left|\widetilde{Q}(0)\right|$. Therefore, $\widetilde{Q}(z_1)$ fits the requirements of Theorem 5.1 of [4], which can be used to show that

$$\max_{z_1 \in [1-c_1, 1-c_2]} \widetilde{Q}(z_1) \geq \exp\left(Cn^\mu \log^6 n\right) \tag{9}$$

Combining equations (8) and (9) yields our claim. ◀

## 3.2 Completing the Proof

In Section 3.1, we proved that the estimation method promised in Theorem 7 can be used to differentiate between any two potential input strings $\mathbf{x}$ and $\mathbf{y}$ from their traces. In this section, we will show how this distinguishing oracle can be transformed into a reconstruction algorithm, completing the proof of Theorem 2.

The basic idea of this transformation is relatively simple. We enumerate over potential pairs of input strings $\mathbf{y}^0, \mathbf{y}^1$, and use the distinguishing oracle to decide for each pair which is a better candidate for being the input string $\mathbf{x}$.

The main technical difficult we need to overcome is due to the fact that the input string $\mathbf{x}$ may be arbitrarily long, so enumerating over all possible input strings can take an arbitrarily long amount of time. We overcome this, by showing that it suffices to enumerate over the first $O(n)$ bits of the input string. Moreover, when the deletion probability is below $1/2$, we show that it suffices to enumerate over only a small fraction of the entropy of these $O(n)$ bits, yielding a fast reconstruction algorithm.

Let $\mathbf{x}$ be the input string to the shifted trace reconstruction problem. By our definition of the shifted trace reconstruction problem, the first $n$ bits of $\mathbf{x}$ are known, and our goal is to reconstruct the $n+1$th bit of the input string $\mathbf{x}$.

Let $C > 0$ be a sufficiently large constant. Let $\mathbf{o}^0, \mathbf{o}^1 \in \{0,1\}^{Cn-n-1}$ be two hypotheses for the value of $\mathbf{x}_{n+1:Cn}$. In other words, $\mathbf{y}^0 = \mathbf{x}_{1:n}0\mathbf{o}^0, \mathbf{y}^1 = \mathbf{x}_{1:n}1\mathbf{o}^1$ are our hypotheses for the first $Cn$ bits of $\mathbf{x}$.

If $\delta < 1/2$, let $z_0$ and $\mathbf{w}$ be as defined in Corollary 11, and let $z_1 = 0$. If $\delta \geq 1/2$, let $z_0, z_1$ and $\mathbf{w}$ be as defined in Corollary 12.

We use the traces to estimate $p_{I_\mathbf{w},\mathbf{x}}(z_0, z_1, \ldots, z_1)$ using the method promised by Theorem 7. This method may have a small failure probability (which would result in a bad estimate), but for the moment we assume that it succeeds. We then compute $p_{I_\mathbf{w},\mathbf{y}}(z_0, z_1, \ldots, z_1)$ directly for $\mathbf{y} \in \{\mathbf{y}^0, \mathbf{y}^1\}$.

Consider the case where $\mathbf{y}^b = \mathbf{x}_{:Cn}$ for some $b \in \{0,1\}$. Because we are evaluating $p_{I_\mathbf{w},\mathbf{y}}$ at points with coordinates strictly below 1 in absolute value and this polynomial's coefficients are bounded by 1, the contribution of monomials with total degree above $Cn$ is can be bounded. In particular,

$$\begin{aligned} \left|p_{I_\mathbf{w},\mathbf{y}^b}(z_0, z_1, \ldots, z_1) - p_{I_\mathbf{w},\mathbf{x}}(z_0, z_1, \ldots, z_1)\right| &< \exp(-\Omega(Cn^{1/5}\log^6 n)) \ll \\ \ll \left|p_{I_\mathbf{w},\mathbf{y}^b}(z_0, z_1, \ldots, z_1) - p_{I_\mathbf{w},\mathbf{y}^{1-b}}(z_0, z_1, \ldots, z_1)\right| \end{aligned} \tag{10}$$

Therefore, in this case, our estimate of $p_{I_\mathbf{w},\mathbf{x}}(z_0, z_1, \ldots, z_1)$ from the traces will be closer to $p_{I_\mathbf{w},\mathbf{y}^b}(z_0, z_1, \ldots, z_1)$ than to $p_{I_\mathbf{w},\mathbf{y}^{1-b}}(z_0, z_1, \ldots, z_1)$.

We repeat this process for any such pair $\mathbf{o}^0, \mathbf{o}^1 \in \{0,1\}^{Cn-n-1}$, and for each such pair, we output the value $b$ for which our estimate of $p_{I_\mathbf{w},\mathbf{x}}(z_0, z_1, \ldots, z_1)$ from the traces is closest to $p_{I_\mathbf{w},\mathbf{y}^b}(z_0, z_1, \ldots, z_1)$.

If $b = \mathbf{x}_{n+1}$, then there exists at least one such $\mathbf{o}^b$ for which the process above always selects $b$ for any $\mathbf{o}^{1-b}$. By enumerating over all pairs, we can find the value of $b = \mathbf{x}_{n+1}$ for which such a string $\mathbf{o}^b$ exists.

This leaves only a few minor technical details in order to prove Theorem 2.

First, we note that the estimation oracle promised in Theorem 7 has a small failure probability. We use the union bound to show that the probability that it will fail even once in the process described above is negligible.

Next we consider the time complexity of our reconstruction. For the high deletion probability regime ($\delta \geq 1/2$), this process can clearly be completed in time $\exp(O(n))$.

For lower deletion probabilities $\delta < 1/2$, we note that $p_{I_\mathbf{w},\mathbf{y}}(z_0, 0, \ldots, 0)$ depends only on the indices within $\mathbf{y}$ where the string $\mathbf{w}$ appears as a consecutive substring. By our definition of $\mathbf{w}$ (see the proof of Corollary 11), this set of indices is sparse. By enumerating only over the set of indices where $\mathbf{w}$ appears in $\mathbf{y}$ (and not over the entire $Cn$ bits), we can reduce the time complexity of this reconstruction algorithm to $\exp(o(n))$, thus completing our proof of Theorem 2.

## 4 Proof of Theorem 4

In Sections 2 and 3, we showed that Chase's worst-case trace reconstruction method can be naturally extended to the shift trace reconstruction problem and to SID channels. In this section, we will construct a general reduction from the average-case trace reconstruction problem to the shifted trace reconstruction problem, proving Theorems 4 and 6.

Our proof will be based based on an adaptation of the HPPZ's methods, and our main contribution is to show that it can be used as a general reduction as well as to extend it to symmetry channels. Due to space limitations, in this version of the paper we will give only a sketch of the proof (for more details, see the full version of this paper [23])

Our reduction will consist of three main ingredients:

- A Boolean test $T(\mathbf{w}, \widetilde{\mathbf{w}})$ on pairs of bit-strings $(\mathbf{w}, \widetilde{\mathbf{w}})$ that returns 1 if $\widetilde{\mathbf{w}}$ is a plausible match for the output of applying the channel $\mathcal{C}$ to $\mathbf{w}$.
- A two-step alignment procedure comprised of a coarse and a fine alignment each of which uses the test $T$ to obtain an estimate $\tau^k$ for the positions within some of the traces corresponding to the $k$th bit of the original message $\mathbf{x}$.
- The reduction target – a bit recovery procedure based on the target of our reduction to produce an estimate of any bit of $\mathbf{x}$ from these aligned traces.

Finally, similar to HPPZ, throughout this section we will perform our analysis when $\delta = \sigma$, but all of these results can be similarly generalized for any values of $\delta, \sigma \in [0, 1)$.

## 4.1 The Boolean Test

The first component of our reduction is a Boolean test $T$ designed to answer whether a string $\widetilde{\mathbf{w}}$ is likely to have originated from a trace of some string $\mathbf{w}$ or not.

Let $\ell, \lambda < \sqrt{\ell}$ and $c \in (0, 1)$ be parameters of the test. The test $T^c_{\ell,\lambda}$ (when $c, \ell, \lambda$ are clear from the context we may omit them) is defined as follows. First, each of the strings $\mathbf{w}$ and $\widetilde{\mathbf{w}}$ is split into $\approx \ell/\lambda$ segments of length $\lambda$ each. Each segment of each string is assigned a sign $+1$ if most of the bits in this segment are 0s or $-1$ otherwise. In other words

$$s_i = \operatorname{sign}\left\{ \sum_{i\lambda < j \le (i+1)\lambda} (\mathbf{w}_j - 1/2) \right\} \in \{\pm 1\}.$$

Then, the signs of the segments are compared, and we compute the number of segment pairs whose signs agree. If $\mathbf{w}$ and $\widetilde{\mathbf{w}}$ were two independently distributed random strings, then the number of such pairs would be distributed according to the $\mathrm{Bin}(1/2, \ell/\lambda)$ distribution. If $\mathbf{w}$ and $\widetilde{\mathbf{w}}$ are similar, then we expect these signs to be roughly correlated to one another.

Therefore, we define the test to pass if at least $(1 + c)/2$ fraction of the signs agree.

$$T^c_{\ell,\lambda} = \begin{cases} 1 & \sum_{1 \le i \le \ell/\lambda} s_i \widetilde{s}_i > c \\ 0 & \text{otherwise} \end{cases}$$

We consider this test with two sets of parameters for "coarse" and "fine" alignment procedures. Let $H(n) = \exp(h(n))$ be the sample complexity of the shifted trace reconstruction problem. Then for the coarse and fine alignments we set respectively

$$\ell_c = \Theta\left(\frac{\log^2(n)}{h(\Theta(\log(n)))}\right); \qquad \lambda_c = \Theta\left(\frac{\log(n)}{h(\Theta(\log(n)))}\right)$$
$$\ell_f = \Theta\left(h(\log(n))\right); \qquad \lambda_f = \Theta(1).$$

Ideally, we want this Boolean test to maintain two behaviours:

- If $\widetilde{\mathbf{w}}$ is not a trace of $\mathbf{w}$, the probability that $T$ will return 1 (called a *spurious match*) should be at most $\exp(-\Omega(\ell/\lambda))$.
- If $\widetilde{\mathbf{w}}$ is a trace of $\mathbf{w}$, the probability that $T$ will pass (called a *true match*) will be at least $\exp(-O(\ell/\lambda^2))$.

If these conditions hold, then the probability of a true match may be very small, but when $\lambda$ is sufficiently large, it will be much higher than the probability of a spurious match. Therefore, when conditioning on a match, it will most likely be a true match. Over the next few paragraphs, we will give a sketch of the proof that these conditions hold for substrings of a random string $\mathbf{x}$.

### 4.1.1 Spurious Matches are Rare

If $\mathbf{w}$ and $\widetilde{\mathbf{w}}$ are two independently distributed strings chosen uniformly at random, then the signs of their segments $s_i$ as defined above will also be independent and uniformly distributed vectors $s, \widetilde{s} \in \{\pm 1\}^{\ell/\lambda}$. In this case, it can be easily shown from the Chernoff bound that the probability that more than $(1+c)/2$ fraction of their entries agree decays exponentially in their dimension $\ell/\lambda$.

The main difficulty is analysing how this relates to the traces of a random string. Let $\mathbf{w}^0 = \mathbf{x}_{a_0:b_0}$ and $\mathbf{w}^1 = \mathbf{x}_{a_1:b_1}$ be two substrings of the random input string $\mathbf{x}$. If the segments $[a_0, b_0]$ and $[a_1, b_1]$ do not overlap, then (averaging over the random options for the input string $\mathbf{x}$) they are two independent random strings.

Let $\widetilde{\mathbf{w}}^i$ be the trace of $\mathbf{w}^i$. Clearly, when applying the channel $\mathcal{C}$ (which only deletes bits, inserts i.i.d. uniformly distributed bits and replaces some of the bits of $\mathbf{x}$ with i.i.d. uniformly distributed bits) to a random string of length $\ell$, the output will also be a random string of length roughly $\frac{1-\delta}{1-\sigma}\ell = \ell$. Therefore, if $\mathbf{w}^0, \mathbf{w}^1$ are non-overlapping substrings of $\mathbf{x}$ as defined above, then $\mathbf{w}^0$ and $\widetilde{\mathbf{w}}^1$ are two independent random strings.

Let us denote by $\omega$ the randomness of the channel $\mathcal{C}$. Averaging over both the randomness of the channel and over our selection of the input string $\mathbf{x}$, we have

$$\begin{aligned}\mathop{\mathbb{E}}_{\mathbf{x}}\left[\Pr_{\omega}\left[T(\mathbf{w}^0, \widetilde{\mathbf{w}}^1) = 1\right]\right] &= \Pr_{\omega,\mathbf{x}}\left[T(\mathbf{w}^0, \widetilde{\mathbf{w}}^1) = 1\right] \\ &= \Pr_{\mathbf{w}^0,\mathbf{w}^1 \leftarrow \{0,1\}^\ell}\left[T(\mathbf{w}^0, \mathbf{w}^1)\right] = \exp\left(-\Omega(\ell/\lambda)\right)\end{aligned} \tag{11}$$

We will use equation (11) in the two settings of the alignment procedure. In the coarse alignment, we set $\ell_c/\lambda_c = C\log(n) = \Theta(\log(n))$. Setting $C$ to be sufficiently large, we can ensure that $\Pr_{\omega,\mathbf{x}}\left[T(\mathbf{w}^0, \widetilde{\mathbf{w}}^1) = 1\right] = \exp(-\Omega(C\log(n))) < n^{-10}$ is sufficiently small that a simple union bound on the quasi-linear number of coarse alignment procedures we run will never result in a spurious match.

For the fine alignment procedure, we will have a segment $I = [a, a + C\log(n)]$ of length $\Theta(\log n)$ of the input string $\mathbf{x}$ in which our goal will be to find a subsegment $S = [b, b+\ell_f]$ of length $\ell_f = o(\log(n))$ such that for any non-overlapping subsegment $S' \subset I$ of length $\ell_f$, the probability of a spurious match between $\mathbf{w}^0 = \mathbf{x}_S$ and a trace of $\mathbf{w}^1 = \mathbf{x}_{S'}$ is

$$\Pr_{\omega}\left[T(\mathbf{w}^0, \widetilde{\mathbf{w}}^1) = 1\right] = \exp(-\Omega(\ell_f/\lambda_f)).$$

It can be shown from a simple combination of a Markov inequality (used to show that the probability of any such subsegment $S$ to work is $1 - \exp(-\Omega(\ell_f)))$ and an enumeration over sufficiently many independent options for $S$, that at least 1 such subsegment exists w.p. $1 - \exp(-\Omega(C\log(n))) = 1 - n^{-10}$. From here, we can simply apply the union bound over the quasi-linear number of fine alignment procedures in the reduction.

### 4.1.2 True Matches are Frequent

The next step of our proof will be to show that a string $\mathbf{w}$ and its trace $\widetilde{\mathbf{w}}$ will pass the test $T_{\ell,\lambda}$ with probability at least $\exp\left(-O(\ell/\lambda^2)\right)$. Due to space limitations, we give only a very rough sketch of this proof (for a more detailed proof, see the full version of this paper [23]).

Consider a substring $\mathbf{u} = \mathbf{w}_{i\lambda:(i+1)\lambda}$ of the string $\mathbf{w}$ the matching substring $\widetilde{\mathbf{w}}_{i\lambda:(i+1)\lambda}$ of its trace. The total of the bits in $\mathbf{u}$ is binomially distributed, so there is a non-negligible probability that $\approx 1/2 + \sqrt{1/\lambda}$ fraction of them will be 0 (in which case, its sign will be $-1$). If this is the case, then with fairly high probability, for any substring $\mathbf{u'} = \mathbf{w}_{i\lambda+d_i,(i+1)\lambda d_{i+1}}$ where $|d_i| < \lambda/100$, at least $\approx 1/2 + \sqrt{1/2\lambda}$ fraction of its bits will be 0.

For now, assume that $\widetilde{\mathbf{w}}_{i\lambda:(i+1)\lambda} = \widetilde{\mathbf{u'}}$ originated from the application of the channel $\mathcal{C}$ to $\mathbf{u'}$. The channel replaced a constant fraction of the bits of $\mathbf{u'}$ with random bits (through the symmetry portion of the channel or the insertion and deletion portions). However, a constant fraction of these bits were retained, so there is some correlation between their total and that of the string $\mathbf{u'}$. It can be shown that this correlation suffices to ensure a probability of at least $1/2 + \Omega(1)$ that the sign of this segment $\widetilde{s}_i$ of the trace will be equal to the sign of the appropriate segment $s_i$ of the input string $\mathbf{w}$.

These correlations suffice to ensure that on average $1/2 + \Omega(1)$ of the segments of the trace $\widetilde{\mathbf{w}}$ of an input string $\mathbf{w}$ will have the same sign as the appropriate segments of the input string $\mathbf{w}$, conditioned on each of the *mismatches* $d_i$ being at most $|d_i| < \lambda/100$ (with probability $1 - \exp(-\Omega(\ell))$ over the choice of $\mathbf{w}$). Therefore, if we properly set the constant $c$ parameter of the test $T$, under these conditions the probability of a true match will be at least $\Omega(1)$.

The next step of our analysis is to show that the mismatches $d_i$ are sufficiently small with probability at least $\exp(-O(\ell/\lambda^2))$. A formal version of this analysis can be found in the full version of our paper [23].

## 4.2 Coarse and Fine Alignments

Next, we define our coarse and fine alignment procedures. Let be $C$ a sufficiently large constant. We define the parameters for the test used in our coarse and fine alignment procedures to be:

$$\begin{aligned} \ell_c &= C\frac{\log^2 n}{h(C\log n)}; \qquad \lambda_c = C^{1/2}\frac{\log n}{h(C\log n)} \\ \ell_f &= C^{2/3}h(C\log n); \qquad \lambda_f = C^{1/12} \end{aligned}$$

In the full version of this paper [23], we define a precise condition on the input string $\mathbf{x}$ being "well-behaved" (denoted by $\mathbf{x} \in \Xi_{\text{good}}$), and show that a string $\mathbf{x} \in \{0,1\}^n$ selected uniformly at random is well-behaved with probability $1 - n^{-2}$. We define our alignment procedure for well-behaved strings $\mathbf{x}$.

Let $\mathbf{x} \in \Xi_{\text{good}}$ be a well-behaved string. For any integer $k \in [\ell_c + C\log n, n]$, we set the index $a_1 = k - \ell_c - C\log n$ and select $a_2 \in [k - 2/3C\log n, k - 1/3C]$ through a process defined in the full version of this paper [23].

For any trace $\widetilde{\mathbf{x}}$, we set our *coarse alignment* $\tau_1^k$ to be the first integer $b$ for which

$$T_{\ell_c,\lambda_c}(\mathbf{x}([a_1, a_1 + \ell_c]), \widetilde{\mathbf{x}}([b, b + \ell_c])) = 1$$

or $\infty$ if no such $b$ exists. For any trace $\widetilde{\mathbf{x}}$ with $\tau_1^k < \infty$, we define its *fine alignment* $\tau_2^k$ to be the first index $b \in [\tau_1^k - \ell_c, \tau_1^k + 2\ell_c + C\log n]$ such that

$$T_{\ell_f,\lambda_f}(\mathbf{x}([a_2, a_2 + \ell_f]), \widetilde{\mathbf{x}}([b, b + \ell_f])) = 1.$$

We define the *mismatch* $d(k,\tau_i^k)$ of any finite alignment $\tau_i^k<\infty$ as the distance between $\tau_i^k$ and the index of the first bit of the trace originating from the $k$th bit of the input message onwards $\mathbf{x}_{k:}$. The following lemma (which we prove in the full version of this paper [23]) promises that $\tau_i^k<\infty$ with sufficiently high probability and that there is a negligible probability that the mismatch of $\tau_i^k$ is large.

▶ **Lemma 13.** *Let $\boldsymbol{x}\in\Xi_{good}$ be a well-behaved string and let $k\in\{\ell_c+C\log n,\dots,n\}$ be an integer. Then for $a_1,a_2,\tau_1^k,\tau_2^k$ as defined above, the following properties hold:*

- $\Pr\left[\tau_1^k<\infty\right]>\exp(-c_1C^{1/2}h(C\log n))$
- $\Pr\left[\tau_1^k<\infty\wedge d(k,\tau_1^k)>\ell_c\right]<n^{-2}$
- $\Pr\left[\tau_2^k<\infty\mid\tau_1^k<\infty\right]\geq\exp(-c_2C^{1/2}h(C\log n))$
- $\Pr\left[\tau_2^k<\infty\wedge d(k,\tau_2^k)>\ell_f\mid\tau_1^k<\infty\right]<\exp(-c_3C^{7/12}h(C\log n))$

*Where the probabilities are taken over the randomness of the channel and $c_1,c_2,c_3,c_4>0$ are positive constants that may depend on $\delta,\sigma,\gamma$ but not on $C$ or $n$ and originate from the $\Omega(\cdot)$s and $O(\cdot)$s of the previous sections.*

Moreover, as we prove in the full version of this paper, this alignment can be performed efficiently.

▶ **Lemma 14** ($\tau_1^k,\tau_2^k$ **can be computed efficiently).** *There is an algorithm $A_{\text{align}}$ such that, for any $\boldsymbol{x}\in\Xi_{good}$, $k\in\{\ell_c+C\log n,\dots,n\}$ and any trace $\widetilde{\boldsymbol{x}}$ of $\boldsymbol{x}$ through the channel, given*

$$k,\boldsymbol{x}_{:k},(\tau_1^1,\dots,\tau_1^{k-1}),(\tau_2^1,\dots,\tau_2^k)$$

*$A_{\text{align}}$ computes $\tau_1^k,\tau_2^k,a_2$ in time $n^{o(1)}$, with probability $\geq 1-n^{-2}$.*

## 4.3 Using the Oracle

In Section 4.1, we introduced the Boolean test which can be used to test whether a substring of a trace $\widetilde{\mathbf{x}}$ originated from a specific substring of the input string $\mathbf{x}$. Then, in Section 4.2, we showed that this test can be used as a central component of an alignment procedure which maps indices of the input string $\mathbf{x}$ to their positions in the traces $\widetilde{\mathbf{x}}$ with high probability. In this section, we will complete the proof of our reduction from the average-case trace reconstruction problem to the shifted trace reconstruction problem.

**Proof of Theorem 4.** Let $\mathcal{C}$ be an SID channel with parameters $\gamma,\sigma,\delta$, and let $C$ to be a sufficiently large constant.

We will prove that given the first $k\geq\ell_c+C\log n$ bits of $\mathbf{x}$, we can reconstruct the rest of its bits one at a time. We can work under this assumption, by adding $\ell_c+C\log n$ virtual 0 bits to the start of $\mathbf{x}$ and adding a trace of $0^k$ to the beginning of each of the traces $\widetilde{\mathbf{x}}$ before the reconstruction.

Given the first $k$ bits of $\mathbf{x}$, we will show that we can reconstruct the $k+1$th bit of $\mathbf{x}$ and from there, we can continue this process iteratively. Using the alignment algorithm from Lemma 14, we compute $\tau_1^k$ and $\tau_2^k$ of each of the traces $\widetilde{\mathbf{x}}$.

Given $a_2,\tau_2^k$, we run the shifted trace reconstruction algorithm $A$ with parameters $n',n'-1$, where $n'=k-a_2\in[1/3C\log n,2/3C\log n]$, on the set:

$$\mathcal{X}=\left\{\widetilde{\mathbf{x}}(\tau_2^k:)\middle|_{\tau_2^k(\widetilde{\mathbf{x}})<\infty}^{\widetilde{\mathbf{x}}\text{ is a sample}}\right\}$$

The first and third claims of Lemma 13, mean that for each of our $N=\exp(Ch(C\log n))$ traces, it will have a finite $\tau_2^k$, with probability at least

$$\exp(-C^{1/2}(c_1+c_2)h(C\log n))\geq\exp(-1/3Ch(C\log n)).$$

Therefore, by Hoeffding's inequality, the probability that we will have at least

$$1/2\exp(2/3Ch(C\log n)) > \exp(1/2Ch(2/3C\log n)) \geq \exp(h(k-a_2))\log^2(n)$$

traces for which $\tau_2^k < \infty$ is at least

$$1-\exp(-\Omega(Ch(C\log n))) = 1-n^{-\omega(1)}$$

Lemma 13 gives us that the probability that any sample for which $\tau_2^k < \infty$ is the result of a spurious match is at most

$$\varepsilon(n) \leq \exp(-(C^{7/12}c_3 - C^{1/2}(c_1+c_2))h(C\log n)) \leq \exp(-10h(k-a_2))$$

Splitting our samples into $\log^2(n)$ batches of size $\exp(h(k-a_2))$ each, we ensure that

1. From the union bound, for each batch, the probability that even a single sample is due to a spurious match is at most $\exp(-9h(k-a_2)) = o(1)$.
2. For each batch, if this batch contained no spurious matches, then applying the shifted trace reconstruction oracle on this batch separately will yield the correct value of the bit $\mathbf{x}_k$ with probability $1-o(1)$.
3. The batches are independent of one another.

From here we can use to Chernoff bound to show that the probability that more than $1/3$ of these batches either has at least one spurious match or yielded the wrong output from the shifted trace reconstruction oracle is $\exp(-\Omega(\log^2(n))) = n^{-\omega(1)}$, so taking a majority vote on the applications of the shifted trace reconstruction oracle will yield the correct value of $\mathbf{x}$ with probability $1-n^{-\omega(1)}$, completing our proof. ◀

## 5 Conclusions

In this paper we presented two main results. First, we proved a general reduction from the average-case trace reconstruction problem to the shifted trace reconstruction problem, which is similar to the worst-case trace reconstruction problem. Second, we generalised the leading algorithm for the worst-case trace reconstruction problem from deletion channels by Chase [7] to the shifted trace reconstruction problem and to the more general class of symmetry-insertion-deletion channels.

Our reduction is based on the work of Holden et al. [14] who used a similar technique to convert the specific methods of De et al. and Nazarov and Peres [12, 20] from worst-case trace reconstruction to the average-case. Continuing the line of work of Brakensiek et al. [5] who reduced the coded trace reconstruction problem to the average-case trace reconstruction problem, we convert the specific construction of Holden et al. to a reduction. Altogether a computational class of trace reconstruction problems begins to emerge.

Moreover, we note McGregor et al. [17] whose results prove that up to the differences between shifted and worst-case trace reconstruction, our reduction is essentially tight. This leads us to several interesting possibilities for future research on trace reconstruction.

First, many other versions of the trace reconstruction have been introduced over the last few years and analysed with an extension of the methods of De et al. and Nazarov and Peres [12, 20] for worst-case trace. If more of these analyses can be converted to reductions to the worst-case or average-case trace reconstruction problems, this would help to simplify the analysis of the many open questions in this field.

Secondly, it seems that the best known techniques for the worst-case trace reconstruction problem translate nicely to the shifted trace reconstruction problem, leading to the conjecture that the two are equivalent. A reduction between the two would help focus further research on this problem.

Finally, we note our extension of Chase's analysis to symmetry-insertion-deletion channels. This portion of our proof is complicated and would be difficult to extend to other settings. An important question for future research is whether there exists a simpler and more elegant analysis for these channels.

## References

1 Alexandr Andoni, Constantinos Daskalakis, Avinatan Hassidim, and Sebastien Roch. Global alignment of molecular sequences via ancestral state reconstruction. *Stochastic Processes and their Applications*, 122(12):3852–3874, 2012.

2 Tugkan Batu, Sampath Kannan, Sanjeev Khanna, and Andrew McGregor. Reconstructing strings from random traces. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '04, pages 910–918, USA, 2004. Society for Industrial and Applied Mathematics.

3 Peter Borwein and Tamás Erdélyi. Littlewood-type problems on subarcs of the unit circle. *Indiana University mathematics journal*, pages 1323–1346, 1997.

4 Peter Borwein, Tamás Erdélyi, and Géza Kós. Littlewood-type problems on [0, 1]. *Proceedings of the London Mathematical Society*, 79(1):22–46, 1999.

5 Joshua Brakensiek, Ray Li, and Bruce Spang. Coded trace reconstruction in a constant number of traces. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 482–493. IEEE, 2020.

6 Zachary Chase. New lower bounds for trace reconstruction. In *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, pages 627–643. Institut Henri Poincaré, 2021.

7 Zachary Chase. Separating words and trace reconstruction. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 21–31, 2021.

8 Zachary Chase and Yuval Peres. Approximate trace reconstruction of random strings from a constant number of traces. *arXiv preprint*, 2021. `arXiv:2107.06454`.

9 Xi Chen, Anindya De, Chin Ho Lee, Rocco A Servedio, and Sandip Sinha. Polynomial-time trace reconstruction in the smoothed complexity model. *ACM Transactions on Algorithms (TALG)*, 2020.

10 Mahdi Cheraghchi, Ryan Gabrys, Olgica Milenkovic, and Joao Ribeiro. Coded trace reconstruction. *IEEE Transactions on Information Theory*, 66(10):6084–6103, 2020.

11 Sami Davies, Miklós Z Rácz, Benjamin G Schiffer, and Cyrus Rashtchian. Approximate trace reconstruction: Algorithms. In *2021 IEEE International Symposium on Information Theory (ISIT)*, pages 2525–2530. IEEE, 2021.

12 Anindya De, Ryan O'Donnell, and Rocco A Servedio. Optimal mean-based algorithms for trace reconstruction. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1047–1056, 2017.

13 Nina Holden, Robin Pemantle, Yuval Peres, and Alex Zhai. Subpolynomial trace reconstruction for random strings and arbitrary deletion probability. In *Conference On Learning Theory*, pages 1799–1840. PMLR, 2018.

14 Nina Holden, Robin Pemantle, Yuval Peres, and Alex Zhai. Subpolynomial trace reconstruction for random strings and arbitrary deletion probability. *Mathematical Statistics and Learning*, 2(3):275–309, 2020.

15 Thomas Holenstein, Michael Mitzenmacher, Rina Panigrahy, and Udi Wieder. Trace reconstruction with constant deletion probability and related results. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 389–398, 2008.

16 Akshay Krishnamurthy, Arya Mazumdar, Andrew McGregor, and Soumyabrata Pal. Trace reconstruction: Generalized and parameterized. *IEEE Transactions on Information Theory*, 67(6):3233–3250, 2021.

17 Andrew McGregor, Eric Price, and Sofya Vorotnikova. Trace reconstruction revisited. In *European Symposium on Algorithms*, pages 689–700. Springer, 2014.

18 Michael Mitzenmacher. A survey of results for deletion channels and related synchronization channels. *Probability Surveys*, 6:1–33, 2009. doi:10.1214/08-PS141.

19 Shyam Narayanan and Michael Ren. Circular trace reconstruction. *arXiv preprint*, 2020. arXiv:2009.01346.

20 Fedor Nazarov and Yuval Peres. Trace reconstruction with exp (o (n1/3)) samples. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1042–1046, 2017.

21 Yuval Peres and Alex Zhai. Average-case reconstruction for the deletion channel: subpolynomially many traces suffice. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 228–239. IEEE, 2017.

22 John M Robson. Separating strings with small automata. *Information processing letters*, 30(4):209–214, 1989.

23 Ittai Rubinstein. Average-case to (shifted) worst-case reduction for the trace reconstruction problem. *arXiv preprint*, 2022. arXiv:2207.11489.