Private and Resource-Bounded Locally Decodable Codes for Insertions and Deletions[*]

Alexander R. Block and Jeremiah Blocki

Purdue University

Email: {block9, jblocki}@purdue.edu[†]

Abstract—We construct locally decodable codes (LDCs) to correct insertion-deletion errors in the setting where the sender and receiver share a secret key or where the channel is resource-bounded. Our constructions rely on a so-called "Hamming-to-InsDel" compiler (Ostrovsky and Paskin-Cherniavsky, ITS '15 & Block et al., FSTTCS '20), which compiles any locally decodable Hamming code into a locally decodable code resilient to insertion-deletion (InsDel) errors. While the compilers were designed for the classical coding setting, we show that the compilers still work in a secret key or resource-bounded setting. Applying our results to the private key Hamming LDC of Ostrovsky, Pandey, and Sahai (ICALP '07), we obtain a private key InsDel LDC with constant rate and polylogarithmic locality. Applying our results to the construction of Blocki, Kulkarni, and Zhou (ITC '20), we obtain similar results for resource-bounded channels; i.e., a channel where computation is constrained by resources such as space or time.

I. INTRODUCTION

Error-correcting codes that are resilient to insertion-deletion (InsDel) errors have been a major focus in recent advances in coding theory [Lev66, KLM04, GW17, HS17, GL19, GL18, HSS18, HS18, BGZ18, CJLW18, CHL+19, CJLW19, HRS19, Hae19, CGHL20, CL20, GHS20, LTX20, SB19]. Such codes are a generalization of classical Hamming codes to handle the case where symbols at arbitrary positions in the codeword can be inserted or deleted. Insertion-deletion codes over alphabet Σ are described by an encoding function $\mathsf{Enc}\colon \Sigma^k \to \Sigma^K$ and decoding function $\mathsf{Dec}\colon \Sigma^* \to \Sigma^k$ such that for a message $x \in \Sigma^k$, if $y' \in \Sigma^*$ such that the edit distance between $\mathsf{Enc}(x)$ and y' is at most $2\rho K$, then $\mathsf{Dec}(y') = x$. A core research direction is building codes with high information rate k/K that are robust to a large constant fraction ρ of insertion-deletion errors. Only recently have *efficient* (i.e., polynomial time encoding and decoding) InsDel codes with asymptotically good (i.e., constant) information rate and error tolerance been well-understood [GHS20, Hae19, HRS19, HS18, LTX20].

Even less understood are *locally decodable* codes for insertions and deletions: such error-correcting codes admit super efficient (e.g., polylogarithmic time) decoding algorithms which, by querying few locations into a received word, can recover portions of the original message. Inspired by locally decodable codes (LDCs) for Hamming errors [KT00, STV99], Ostrovsky and Paskin-Cherniavsky [OPC15] introduced the notion of *locally decodable InsDel codes* (InsDel LDC). A code $C[K, k] = (\mathsf{Enc}, \mathsf{Dec})$ is an (ℓ, ρ, p)-InsDel LDC if the decoding function Dec is a randomized algorithm that makes at most ℓ queries to the received word and, if the edit distance between an encoded message $\mathsf{Enc}(x)$ and a received word y' is at most $2\rho \cdot |\mathsf{Enc}(x)|$, then Dec on input i outputs x_i with probability at least p. Here, ℓ is the *locality* of the code, ρ is the *error rate*, and p is the *success probability*. While LDCs for Hamming errors have been studied for several decades [DGY10, Efr09, KW03, KMRZS17, KS16, Yek08, Yek12], the study of InsDel LDCs is scarce. Besides the results of Ostrovsky and Paskin-Cherniavsky [OPC15] and Block et al. [BBG+20], only Haeupler and Shahrasbi [HS18], to the best of our knowledge, consider locality in the building of synchronization strings, which are an important component of optimal InsDel codes.

Ostrovsky and Paskin-Cherniavsky [OPC15] and Block et al. [BBG+20] both give a so-called "Hamming-to-InsDel" compiler: given any classical Hamming LDC as input, this compiler outputs an InsDel LDC. This reduction preserves the information rate and the error rate of the original Hamming LDC (up to constant factors), and the locality only grows by a polylogarithmic factor (in the length of a codeword). Note this reduction holds for any *classical* Hamming LDC. However, there have been recent advances in examining Hamming LDCs in non-classical [OPS07, CLZ20, BKZ20] or relaxed [BGGZ19] settings. For example, there is a line of work studying Hamming codes in which the channel is computationally bounded [Lip94, MPSW05, GS16, SS16]. In such settings the corruption pattern is selected adversarially by a resource bounded channel (e.g., the channel is probabilistic polynomial time), or has other resource constraints such as space or computation depth (i.e., sequential time), restricting the computations that can be performed. It has been argued that any real world communication channel can be reasonably modeled as a resource-bounded channel [Lip94, BKZ20]. The notion of resource-bounded channels is well-motivated by channels in the real world, which all have some sort of limitations on their computations, and one can reasonably expect error patterns encountered in nature to be modeled by some (not necessarily known) probabilistic polynomial time

algorithm. Thus, the study of Hamming and InsDel codes in non-classical and relaxed settings is well-motivated.

Mirroring the Hamming code results, the non-classical and relaxed settings offer much better tradeoffs than classical LDCs, at the cost of different assumptions in the adversarial models, or by allowing the decoder to fail on a small fraction of inputs. For example, codes constructed using secret-key cryptography (i.e., the encoder and decoder share a secret key) admit constant-rate Hamming LDCs with polylogarithmic locality (in the security parameter) [OPS07]. Similarly, when assuming the adversarial channel is resource-constrained in some way (e.g., the channel is a low-depth circuit), there are constructions of constant-rate Hamming LDCs with polylogarithmic locality [BKZ20]. Further, it is not out of the question for a shared secret-key assumption, and it has been argued that resource-constrained adversarial channels can model real-world channels reasonably well [Lip94, BKZ20]. Thus we ask

Can we extend non-classical Hamming LDC*s to the insertion-deletion setting?*

A. Our Results

We answer the question in the affirmative for two classes of non-classical Hamming LDCs. First, we consider *private locally decodable codes* (private LDCs). Introduced by Ostrovsky, Pandey, and Sahai [OPS07], private LDCs leverage cryptographic assumptions to construct locally decodable Hamming codes against probabilistic polynomial time adversaries. In particular, private LDCs leverage a (pseudorandom) *secret key* that is shared between the encoder and the decoder, and assumes that any adversary does not receive this secret key. These codes are additionally parameterized by a security parameter λ and a secret key generation function Gen. Second, we consider Hamming LDCs that are secure against *resource-bounded adversaries*. Blocki, Kulkarni, and Zhou [BKZ20] introduce resource-bounded LDCs as an extension of classical Hamming codes in resource-bounded settings [Lip94, MPSW05, GS16, SS16]. These LDCs are secure against any class of adversaries $\mathbb{C}$ that admit some *safe function* that is uncomputable by any adversary $\mathcal{A} \in \mathbb{C}$. For example, in the (parallel) random oracle model any polynomial time algorithm running in sequential time T provably cannot evaluate the function $\mathsf{H}^{T+1}(\cdot)$ so the function would be a safe function against the class $\mathbb{C}$ of probabilistic polynomial time algorithms with sequential time at most T.

We obtain a binary private InsDel LDC from any private Hamming LDC and a binary resource-bounded InsDel LDC from any resource-bounded Hamming LDC by applying the Hamming-to-InsDel compiler of Block et al. [BBG+20].

Informal Theorem 1 (see Theorem 1): Let $C[K, k]$ be an (ℓ, ρ, p)-private Hamming LDC. There exists a binary $(\ell \cdot \mathrm{polylog}(K), \Theta(\rho), O(p))$-private InsDel LDC with codeword length $\Theta(K)$.

Ostrovsky, Pandey, and Sahai [OPS07] construct a private Hamming LDC over any constant-sized alphabet that achieves constant-rate, $\omega(\log(\lambda))$ locality, constant error rate, and suc-

cess probability $1 - \mathsf{negl}(\lambda)$, where λ is the security parameter and $\mathsf{negl}(\cdot) = o(1/|p(\cdot)|)$ for any non-zero polynomial p. Combining [OPS07] with Informal Theorem 1 yields a constant-rate private InsDel LDC with polylogarithmic locality, constant error rate, and high success probability.

Informal Theorem 2 (see Theorem 2): Let $C[K, k]$ be an (ℓ, ρ, p)-Hamming LDC secure against class $\mathbb{C}$. There exists a binary $(\ell \cdot \mathrm{polylog}(K), \Theta(\rho), O(p))$-InsDel LDC secure against class $\mathbb{C}$ with codeword length $\Theta(K)$.

Blocki, Kulkarni, and Zhou [BKZ20] recently construct a Hamming LDC over any constant-sized alphabet, in the random oracle model (i.e., the encoding and decoding functions make use of a cryptographic hash function), that achieves constant-rate, $\mathrm{polylog}(\lambda)$ locality, constant error rate, and success probability $1 - \mathsf{negl}(\lambda)$, for security parameter λ. In the random oracle model their construction provably yields a secure code for any channel class $\mathbb{C}$ admiting a safe function. Combining [BKZ20] with Informal Theorem 2 yields a constant rate InsDel LDC secure against class $\mathbb{C}$ with polylogarithmic locality, constant error rate, and high success probability.

B. Technical Overview

The key technical component of our constructions is the use of a "Hamming-to-InsDel" compiler [OPC15, BBG+20] which transforms any classical Hamming LDC to an InsDel LDC with polylogarithmic blow-up in the locality. The compiler of Block et al. [BBG+20] is a reproving of Ostrovsky and Paskin-Cherniavsky's result, using different techniques and analysis. For simplicity, we use the compiler of Block et al. in this work, which we refer to as the BBGKZ compiler.

The BBGKZ compiler at its core consists of two functions: Compile and RecoverBit. The function Compile takes as input a codeword $y \in \Sigma^K$ that is resilient to ρ-fraction of Hamming errors and outputs a codeword $Y \in \{0, 1\}^n$ that is resilient to ρ'-fraction of insertion-deletion errors. The compiled encoding function operates as follows: it encodes a message x using the given Hamming LDC to obtain the Hamming codeword y, then it applies the function Compile to y and outputs the final InsDel codeword Y. The function RecoverBit, when given query access to some $Y' \in \{0, 1\}^*$, on input i makes $\mathrm{polylog}(|Y'|)$ queries to Y' and attempts to recover y_i, the ith bit of the Hamming codeword y. The BBGKZ compiler guarantees that if $\mathsf{ED}(Y, Y') \leq \rho'$ then for *most* indices $i \in [K]$, RecoverBit outputs the correct bit y_i with high probability.

The challenge in applying the BBGKZ compiler to a private Hamming LDC or a resource-bounded LDC is that we cannot assume that decoding will be correct for *every* corrupted codeword with small Hamming distance. Instead, we require that the channel cannot produce a codeword which fools the decoding algorithm except with negligible probability. In particular, if y is our encoding of a message x then we say that a corrupted codeword y' *fools* the decoder if:

1) the (Hamming/Edit) distance between y and y' is small; and

2) for some index i, the probability that the local decoder, given oracle access to y', outputs the correct bit x_i is less than p.

The security requirement is that any adversary $\mathcal{A}$ produces such a fooling codeword y' with probability at most ε. The difficulty here is proving that applying the BBGKZ compiler to a private code or resource-bounded code preserves the security of the underlying code. Proving the security of our compiled private/resource-bounded code lies in the algorithm `RecoverBit`: given an adversary $\mathcal{A}$ against the compiled InsDel code, we construct a new adversary $\mathcal{A}'$ against the original Hamming code which does the following:

1) obtains challenge message x and Hamming codeword y;
2) obtains InsDel codeword $Y = \texttt{Compile}(y)$;
3) obtains $Y' \leftarrow \mathcal{A}(x, Y)$; and
4) obtains $y'_j \leftarrow \texttt{RecoverBit}^{Y'}(j)$ for all j.

Applying the key property of `RecoverBit` one can show that the Hamming distance between y and y' is suitably small. Furthermore, if Y' fools our local InsDel decoder then one can argue that (w.h.p.) y' fools our local Hamming decoder. Thus, the compiler transforms secret key Hamming LDCs into secret key InsDel LDCs and resource bounded Hamming LDCs into resource bounded InsDel LDCs. For resource bounded channels, there is another subtlety we must account for. Our Hamming adversary $\mathcal{A}'$ requires *slightly* more resources than the original InsDel adversary $\mathcal{A}$; i.e., we need to run `RecoverBit` for each index j (though this can be accomplished in parallel to minimize computation depth). Thus, to obtain an InsDel LDC secure against the channel class $\mathbb{C}$ we need to start with a Hamming LDC secure against a slightly larger class $\mathbb{C}'$.

C. Related Work

Levenstein [Lev66] initiated the study of codes for insertions and deletions. Since this initiation, there has been a large body of works examining InsDel codes (see surveys [MBT10, Mit08, Slo02]). Recently, [SB19] constructed k-deletion correcting binary codes with optimal redundancy, which was extended to systematic binary codes and q-ary codes in [SGB20b, SGB20a]. This line of work answered long standing open problems in the construction of k-deletion correcting codes with optimal redundancy. Random codes with positive information rate and correcting a large fraction of deletion errors were studied in [KLM04, GW17], and efficiently encodable and decodable codes with constant rate and resilient to a constant fraction of insertion-deletion errors were studied extensively in [SZ99, GW17, HS17, CJLW18, HS18, CHL+19, GL19, BGZ18, CGHL20, CL20, GHS20]. Recently, there has been interest in extending "list-decoding" to the setting of InsDel codes. These codes are resilient to a larger fraction of insertion-deletion errors at the cost of outputting a small list of potential codewords (i.e., the loss of unique decoding) [GHS20, HSS18, LTX20]. Another direction due to Haeupler and Shahrasbi [HS18] involves constructing explicit synchronization strings which can be "locally decoded" in the following sense: each index of the string can be computed using values located at a small number of other indices. These explicit and locally decodable synchronization strings are used to imply near linear time interactive coding schemes for insertion-deletion errors.

Cheng, Li and Zheng [CLZ20] propose the notion of locally decodable codes with randomized encoding, in both the Hamming and edit distance regimes. They study such codes in various settings, including where the encoder and decoder share randomness, or the channel is oblivious to the codeword, and hence adds error patterns non-adaptively. For insertion-deletion errors they obtain codes with $K = O(k)$ or $K = k \cdot \log(k)$ and $\mathrm{polylog}(k)$ locality for message length k.

Blocki, Gandikota, Grigorescu, and Zhou [BGGZ19] construct *relaxed* locally correctable and locally decodable Hamming codes in computationally bounded channels. Here, *local correction* states that a corrupt codeword c' can be corrected to codeword c by only querying c' at a bounded number of locations, and *relaxed* means that the correcting or decoding algorithm is allowed to output the value $\bot$ for a small fraction of inputs. Their construction requires a public parameter setup for a collision-resistant hash function, and they obtain relaxed binary locally correctable and decodable Hamming codes with constant information rate and polylogarithmic locality. Recently, Blocki, Kulkarni, and Zhou [BKZ20] introduced Hamming LDCs that are secure against resource-bounded adversaries, in the random oracle model. Here, they construct codes (in the random oracle model) which are resilient to classes of adversaries $\mathbb{C}$ for which there exists a function f that is uncomputable by any $\mathcal{A} \in \mathbb{C}$. They obtain explicit Hamming LDCs with constant information rate and polylogarithmic locality against various classes $\mathbb{C}$ of resource-bounded adversaries.

II. PRELIMINARIES

We let $\lambda \in \mathbb{N}$ denote the security parameter. For $n \in \mathbb{Z}^+$, we let $[n]$ denote the set $\{1, 2, \ldots, n\}$. A function $\vartheta \colon \mathbb{N} \to \mathbb{R}_{\geq 0}$ is said to be negligible if $\vartheta(n) = o(1/|p(n)|)$ for any fixed non-zero polynomial p. We write PPT as a shorthand for probabilistic polynomial time. For any (randomized) algorithm A, we let $y \leftarrow A(x)$ denote the result of running A on some input x.

We consider the *fractional Hamming distance* and the *fractional Edit Distance* metrics, which we denote by HAM and ED, respectively. For two strings $x, y \in \Sigma^K$ for some K, we define $\mathsf{HAM}(x, y) := |\{i \colon x_i \neq y_i\}_{i \in [K]}|/K$. For two strings $x \in \Sigma^K$ and $y \in \Sigma^*$, we define $\mathsf{ED}(x, y)$ is the minimum number of insertions and deletions required to transform x into y (or vice versa), normalized by $2K$.

Definition 1 (Error-correcting Codes): A *coding scheme* $C[K, k, q_1, q_2] = (\mathsf{Enc}, \mathsf{Dec})$ is a pair of encoding and decoding algorithms $\mathsf{Enc} \colon \Sigma_1^k \to \Sigma_2^K$ and $\mathsf{Dec} \colon \Sigma_2^* \to \Sigma_1^k$, where $|\Sigma_i| = q_i$. A code $C[K, k, q_1, q_2]$ is a (ρ, dist) *error-correcting code* for $\rho \in [0, 1]$ and fractional distance dist if for all $x \in \Sigma_1^k$ and $y \in \Sigma_2^*$ such that $\mathrm{dist}(\mathsf{Enc}(x), y) \leq \rho$, we have that $\mathsf{Dec}(y) = x$. Here, ρ is the *error rate* of C. If $q_1 = q_2$, we

simply denote this by $C[K, k, q_1]$. If $\text{dist} = \text{HAM}$, then C is a *Hamming code*; if $\text{dist} = \text{ED}$, then C is an *insertion-deletion code* (InsDel code).

Definition 2 (Locally Decodable Codes): A coding scheme $C[K, k, q_1, q_2] = (\text{Enc}, \text{Dec})$ is an $(\ell, \rho, p, \text{dist})$-*locally decodable code* (LDC) if for all $x \in \Sigma_1^k$ and $y \in \Sigma_2^*$ such that $\text{dist}(\text{Enc}(x), y) \leq \rho$, the algorithm Dec, with query access to word y, on input index $i \in [k]$, makes at most ℓ queries to y and outputs x_i with probability at least p over the randomness of the decoder. Here, ℓ is the *locality* of C and p is the *success probability*.

Private locally-decodable codes were introduced by Ostrovsky, Pandey, and Sahai [OPS07]. The encoding and decoding algorithms of these codes additionally share a secret key that is hidden from any adversarial channel. Intuitively, these codes ensure that (except with small probability) any channel who does not have the secret key will fail to produce a corrupted codeword y' which fools the local decoder.

Definition 3 (One-Time Private LDC): Let λ be the security parameter. A code $C[K, k, q_1, q_2, \lambda]$ consisting of a tuple of PPT algorithms $(\text{Gen}, \text{Enc}, \text{Dec})$ is a $(\ell, \rho, p, \varepsilon, \text{dist})$-*one time private locally decodable code (private* LDC) if:

- $\text{Gen}(1^\lambda)$ is the key generation algorithm that takes 1^λ as input and outputs a secret key $\text{sk} \in \{0, 1\}^*$, for security parameter λ;
- $\text{Enc} : \Sigma_1^k \times \{0, 1\}^* \to \Sigma_2^K$ is the encoding algorithm that takes as input a message $x \in \Sigma_1^k$ and a secret key sk and outputs a codeword $y \in \Sigma_2^K$; and
- $\text{Dec}^{y'} : \{0, 1\}^{\log k} \times \{0, 1\}^* \to \Sigma_1$ is the decoding algorithm that takes as input index $i \in [k]$ and secret key sk, and is additionally given query access to a corrupted codeword $y' \in \Sigma_2^{K'}$ and outputs $b \in \Sigma_1$ after making at most ℓ queries to y'.

We define a predicate $\text{Fool}(y', \rho, p, \text{sk}, x, y) = 1$ if and only if

1) $\text{dist}(y, y') \leq \rho$; and
2) $\exists i \in [k]$ such that $\Pr[\text{Dec}^{y'}(i, \text{sk}) = x_i] < p$, where the probability is taken over the random coins of Dec.

We require that for all adversaries $\mathcal{A}$ and all $x \in \Sigma_1^k$,

$$\Pr[\text{Fool}(\mathcal{A}(y), \rho, p, \text{sk}, x, y) = 1] \leq \varepsilon,$$

where $y \leftarrow \text{Enc}(x, \text{sk})$ and the probability is taken over the random coins of $\mathcal{A}$ and Gen and Enc (if encoding is randomized).

For all of our code definitions, when $q_2 = 2$ we say that the code is a *binary code*.

A. Codes for Resource-Bounded Channels

Recently, Blocki, Kulkarni, and Zhou [BKZ20] studied error-correcting codes against channels which have some resource bound; e.g., the channel is a low-depth circuit, or is a one-tape Turing machine. Intuitively, these codes ensure that (except with small probability) any adversary with insufficient resources will fail to produce a corrupt codeword y' which fools the local decoder.

Definition 4 ($\mathbb{C}$-secure LDC): A code $C[K, k, q_1, q_2] = (\text{Enc}, \text{Dec})$ is a $(\ell, \rho, p, \varepsilon, \text{dist}, \mathbb{C})$-*locally decodable code against class* $\mathbb{C}$ if Dec takes as input index $i \in [k]$, is additionally given query access to a corrupted codeword $y' \in \Sigma_2^{K'}$, and outputs $b \in \Sigma_1$ after making at most ℓ queries to y'. We define predicate $\text{Fool}(y', \rho, p, x, y) = 1$ if and only if

1) $\text{dist}(y, y') \leq \rho$; and
2) $\exists i \in [k]$ such that $\Pr[\text{Dec}^{y'}(i) = x_1] < p$,

where the probability is taken over the random coins of Dec; otherwise $\text{Fool}(y', \rho, p, x, y) = 0$. We require that for all adversaries $\mathcal{A} \in \mathbb{C}$ and all $x \in \Sigma_1^k$,

$$\Pr[\text{Fool}(\mathcal{A}(y), \rho, p, y) = 1] \leq \varepsilon,$$

where the probability is taken over the random coins of $\mathcal{A}$ and the generation of the codeword $y \leftarrow \text{Enc}(x)$.

B. Hamming-to-InsDel Compiler

Ostrovsky and Paskin-Cherniavsky [OPC15] give a compiler which transforms any Hamming LDC to an InsDel LDC with a polylogarithmic blowup in locality. Block et al. [BBG+20] give another compiler which transforms any Hamming LDC into an InsDel LDC with polylogarithmic blowup in locality, reproving the result of [OPC15] with different techniques and analysis. We use the compiler of Block et al. in this work.

Let $C = (\text{Enc}, \text{Dec})$ be a Hamming LDC. Then the compiler works as follows. The compiled encoder is defined as $\text{Enc}_f(x) := \text{Compile}(\text{Enc}(x))$ for any message x. The decoder Dec_f contains a subroutine RecoverBit which, given query access to some $Y' \in \{0, 1\}^{n'}$, on input index i makes at most $O(\log^4(n'))$ queries and with high probability recovers the ith-bit of c correctly for *most* indices of $c = \text{Enc}(x)$ as long as $\text{ED}(Y, Y')$ is sufficiently small. The decoder Dec_f then runs Dec and simulates oracle access to c by using algorithm RecoverBit. We formally capture the properties of the compiler in the following lemma.

Lemma 1 (Block et al. [BBG+20]): There exist functions Compile and RecoverBit such that for any constant $\rho > 0$ and any Hamming LDC $C[K, k, q_1, q_2] = (\text{Enc}, \text{Dec})$ with locality ℓ, there exists $\rho_f = \Theta(\rho)$ such that for any message x and any c' with $\text{ED}(c', y) \leq \rho_f$ for $y = \text{Compile}(\text{Enc}(x)) \in \{0, 1\}^*$:

1) Dec_f has locality $\ell \cdot O(\log^4(K \cdot \log(q_2)))$ and $|y| = \Theta(K \cdot \log(q_2))$;
2) For $c'' = \text{RecoverBit}^{Y'}(1) \circ \cdots \circ \text{RecoverBit}^{Y'}(K \cdot \log(q_2))$, we have $\Pr[\text{Dec}_f^{c'}(i) = x_i] \geq \Pr[\text{Dec}^{c''}(i) = x_i] - \vartheta_1(K \cdot \log(q_2))$; and
3) if $\text{ED}(c', \text{Enc}_f(x)) \leq \rho_f$ then, except with probability $\vartheta_2(K \cdot \log(q_2))$, $\text{HAM}(c'', \text{Enc}(x)) \leq \rho$.

Here, ϑ_1 and ϑ_2 are fixed negligible functions, Compile is computable in parallel time $\text{polylog}(K)$, and c'' is computable in parallel time $\text{polylog}(K)$.

III. One-Time Private Locally Decodable Codes for Insertion-Deletion Channels

Theorem 1: Let $C[K, k, q_1, q_2, \lambda]$ be a $(\ell, \rho, p, \varepsilon, \mathsf{HAM})$-one time private Hamming LDC for constants $\rho, p > 0$. There exists a binary code $C_f[n, k, q_1, 2]$ that is a $(\ell_f, \rho_f, p_f, \varepsilon_f, \mathsf{ED})$-one time private InsDel LDC, where $\ell_f = \ell \cdot O(\log^4(n))$, $\rho_f = \Theta(\rho)$, $p_f < p$, $\varepsilon_f = \varepsilon/(1 - (p_f/p) - (\vartheta_1(n)/p) - \vartheta_2(n))$, and $n = \Theta(K \cdot \log(q_2))$. Here, ϑ_1, ϑ_2 are fixed negligible functions.

Proof: Let $C[K, k, q_1, q_2, \lambda] = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a $(\ell, \rho, p, \varepsilon, \mathsf{HAM})$-one time private Hamming LDC. We define $\mathsf{Gen}_f(1^\lambda) := \mathsf{Gen}(1^\lambda)$. Then for any message x and secret key sk we define $\mathsf{Enc}_f(x, \mathsf{sk}) := \mathsf{Compile}(\mathsf{Enc}(x, \mathsf{sk}))$. Fixing the secret key sk and applying Lemma 1 to the encoding scheme, we see that Dec_f has locality $\ell \cdot O(\log^4(n))$ and the output length of Enc_f is $n = \Theta(K \log q_2)$ bits. The main challenge is proving the security. Suppose towards contradiction that there exists an adversary $\mathcal{A}_f$ such that $\Pr[\mathsf{Fool}(\mathcal{A}_f(Y), \rho_f, p_f, \mathsf{sk}, x, Y) = 1] > \varepsilon_f$ for $Y \leftarrow \mathsf{Enc}_f(x, \mathsf{sk})$. Then we construct an adversary $\mathcal{A}$ such that $\Pr[\mathsf{Fool}(\mathcal{A}(y), \rho, p, \mathsf{sk}, x, y) = 1] > \varepsilon$ for $y \leftarrow \mathsf{Enc}(x, \mathsf{sk})$. Adversary $\mathcal{A}$ works as follows:

1) $\mathcal{A}$ obtains as input $x, y, \lambda, \rho, p, k$, and K, where $y = \mathsf{Enc}(x, \mathsf{sk})$;
2) $\mathcal{A}$ then obtains $Y = \mathsf{Compile}(y)$; and
3) $\mathcal{A}$ then obtains $Y' \leftarrow \mathcal{A}_f(x, Y, \lambda, \rho_f, p_f, k, n)$.

By assumption $\mathsf{ED}(Y, Y') \leq \rho_f$ and with probability at least ε_f there exists index $i \in [k]$ such that

$$\Pr[\mathsf{Dec}_f^{Y'}(i, \mathsf{sk}) = x_i] < p_f.$$

$\mathcal{A}$ then outputs word

$$y' = \mathsf{RecoverBit}^{Y'}(1) \circ \cdots \circ \mathsf{RecoverBit}^{Y'}(K \cdot \log(q_2)).$$

Suppose that $\mathsf{Fool}(Y', \rho_f, p_f, \mathsf{sk}, x, Y) = 1$. Then we have that $\mathsf{ED}(Y, Y') \leq \rho_f$ and there exists $i \in [k]$ such that

$$\Pr[\mathsf{Dec}_f^{Y'}(i, \mathsf{sk}) = x_i] < p_f.$$

By Lemma 1, we have that $\mathsf{HAM}(y, y') \leq \rho$ with probability at least $1 - \vartheta_2(n)$. By definition of Dec_f and Lemma 1, we have that

$$
\begin{aligned}
p_f &> \Pr[\mathsf{Dec}_f^{Y'}(i, \mathsf{sk}) = x_i] \\
&\geq \Pr[\mathsf{Dec}^{y'}(i, \mathsf{sk}) = x_i] - \vartheta_1(n),
\end{aligned}
$$

where the randomness of the second term is taken over the coins of Dec and the coins used by $\mathsf{RecoverBit}$ to generate y', and the randomness of the first term is taken only over the coins of Dec_f. Define the predicate $B_p(y') = 1$ if and only if $\Pr[\mathsf{Dec}^{y'}(i, \mathsf{sk}) = x_i] < p$, and $B_p(y') = 0$ otherwise, where the probability is taken over Dec's coins. Let $\alpha = \Pr[B_p(y')]$, where the probability is taken over the random coins used to generate y' from Y'. Then we have that

$$\Pr[\mathsf{Dec}^{y'}(i, \mathsf{sk}) = x_i] \geq p(1 - \alpha).$$

This implies that $\alpha > 1 - (p_f/p) - (\vartheta_1(n)/p)$. Now consider two events $\mathcal{F}_{\mathsf{HAM}} = \mathsf{Fool}(y', \rho, p, \mathsf{sk}, x, y)$ and $\mathcal{F}_{\mathsf{ED}} = \mathsf{Fool}(Y', \rho_f, p_f, \mathsf{sk}, x, Y)$. Then

$$\Pr[\mathcal{F}_{\mathsf{HAM}} = 1] \geq \Pr[\mathcal{F}_{\mathsf{ED}} = 1] \cdot \Pr[\mathcal{F}_{\mathsf{HAM}} = 1 | \mathcal{F}_{\mathsf{ED}} = 1].$$

By assumption we have that $\Pr[\mathcal{F}_{\mathsf{ED}} = 1] > \varepsilon_f$. Further, by Definition 3, $\mathcal{F}_{\mathsf{HAM}} = 1$ if and only if $\mathsf{HAM}(y, y') \leq \rho$ and there exists $i \in [k]$ such that $\Pr[\mathsf{Dec}^{y'}(i, \mathsf{sk}) = x_i] < p$. Since $\mathcal{F}_{\mathsf{ED}} = 1$, we have that $\mathsf{ED}(Y, Y') \leq \rho_f$, and thus by Lemma 1 we have that $\mathsf{HAM}(y, y') \leq \rho$ with probability at least $1 - \vartheta_2(n)$. Thus

$$\Pr[\mathcal{F}_{\mathsf{HAM}} = 1 | \mathcal{F}_{\mathsf{ED}} = 1] \geq 1 - \vartheta_2(n) - (1 - \alpha)$$

and $\alpha > 1 - (p_f/p) - (\vartheta_1(n)/p)$. Therefore we have that

$$\Pr[\mathcal{F}_{\mathsf{HAM}} = 1] > \varepsilon_f \cdot (1 - (p_f/p) - (\vartheta_1(n)/p) - \vartheta_2(n)),$$

which is a contradiction since the right hand side of the above equation is equal to ε. ∎

IV. Locally Decodable Codes for Resource-Bounded Insertion-Deletion Channels

To construct LDCs for resource-bounded InsDel channels, we first need to introduce the notion of *closure* between algorithms classes. Let $\mathbb{C}$ be a class of parallel algorithms running in at most sequential time T and maximum space usage S. For any $A \in \mathbb{C}$, let $B = \mathsf{Reduce}(A)$ be a reduction from algorithm A to B. We say the class of algorithms $\mathbb{C}'$ is the *closure of $\mathbb{C}$ with respect to* Reduce if $\mathbb{C}'$ is the minimum class of algorithms such that $\mathsf{Reduce}(A) \in \mathbb{C}'$ for all $A \in \mathbb{C}$.

In our context, for parameter N we define Reduce_N as a sequential time $N \cdot \mathrm{polylog}(N)$ reduction that can be executed in parallel for sequential time $\mathrm{polylog}(N)$. Parallel execution incurs an additional $N \cdot \mathrm{polylog}(N)$ space overhead, and sequential execution incurs an additional $\mathrm{polylog}(N)$ space overhead. Thus, if $\mathbb{C}$ is the class of all parallel PPT algorithms running in sequential time T, then $\overline{\mathbb{C}}(N)$ is some class of parallel PPT algorithms running in time $T + \mathrm{polylog}(N)$.

Theorem 2: Let $\mathbb{C}$ be the class of parallel PPT algorithms running in sequential time T and space S, and let $C[K, k, q_1, q_2] = (\mathsf{Enc}, \mathsf{Dec})$ be a $(\ell, \rho, p, \varepsilon, \mathsf{HAM}, \overline{\mathbb{C}}(n))$-LDC for constants $\rho, p > 0$ and $n = O(K \cdot \log(q_2))$. There exists a binary code $C_f[n, k, q_1, 2]$ that is a $(\ell_f, \rho_f, p_f, \varepsilon_f, \mathsf{ED}, \mathbb{C})$-LDC against class $\mathbb{C}$, where $\ell_f = \ell \cdot O(\log^4(n))$, $\rho_f = \Theta(\rho)$, $p_f < p$, and $\varepsilon_f = \varepsilon/(1 - (p_f/p) - (\vartheta_1(n)/p) - \vartheta_2(n))$. Here, ϑ_1, ϑ_2 are fixed negligible functions.

Proof: The proof follows nearly identically to the proof of Theorem 1; namely, we obtain C_f in an identical manner by using the compiler of Lemma 1 with the code C defined above. The main challenge again is the security proof: given adversary $\mathcal{A}_f \in \mathbb{C}$ such that $\Pr[\mathsf{Fool}(\mathcal{A}(Y), \rho_f, p_f, x, Y) = 1] > \varepsilon_f$ for $Y \leftarrow \mathsf{Enc}_f(x)$, we construct an adversary $\mathcal{A} \in \overline{\mathbb{C}}(n)$ such that $\Pr[\mathsf{Fool}(\mathcal{A}(y), \rho, p, x, y) = 1] > \varepsilon$ for $y \leftarrow \mathsf{Enc}(x)$. Adversary $\mathcal{A}$ is constructed identically as in the proof of Theorem 1, except now the constructed adversary only yields a contradiction if we can show that $\mathcal{A} \in \overline{\mathbb{C}}(n)$. By Lemma 1, we have that $\mathsf{Compile}$ is a

$\text{polylog}(K) = \text{polylog}(n)$ parallel time algorithm, and $y' = \texttt{RecoverBit}^{Y'}(1) \circ \cdots \circ \texttt{RecoverBit}^{Y'}(K \log q_2)$ is computable in $\text{polylog}(n)$ parallel time. Finally, $\texttt{Compile}$ and $\texttt{RecoverBit}$ are run independent of the adversary $\mathcal{A}_\mathsf{f}$, we have that the total parallel time of $\mathcal{A}$ is $T + \text{polylog}(n)$, which implies $\mathcal{A} \in \overline{\mathbb{C}}(n)$, yielding our contradiction. ∎

Remark 1: We focus on a simple reduction, but $\texttt{Reduce}$ can be defined in various different ways, so long as for any $\mathcal{A}_\mathsf{f} \in \mathbb{C}$, it holds that constructed adversary $\mathcal{A} \in \overline{\mathbb{C}}$.

V. EXPLICIT CONSTRUCTIONS

As an application of our main results, we give two explicit constructions.

A. Private InsDel Locally Decodable Code Construc

First, we use Theorem 1 with the one-time private Hamming LDC of Ostrovsky, Pandey, and Sahai [OPS07]. For security parameter λ and fixed negligible functions ϑ_1, ϑ_2, their code has constant-rate, locality $\omega(\log(\lambda))$, constant error-rate, success probability $1 - \vartheta_1(\lambda)$, and security $\varepsilon = \vartheta_2(\lambda)$.

Corollary 1: Let $\ell_\mathsf{f} := \ell_\mathsf{f}(\lambda, n) = \omega(\log(\lambda) \cdot O(\log^4(n)))$. There exists a binary code $C_\mathsf{f}[n, k, q_1, 2, \lambda]$ that is a $(\ell_\mathsf{f}, \rho_\mathsf{f}, p_\mathsf{f}, \varepsilon_\mathsf{f})$-one time private InsDel LDC with constant information rate $k/n = \Theta(1)$, where , $\rho_\mathsf{f} = \Theta(1)$, $p_\mathsf{f} = \Theta(1)$, and $\varepsilon_\mathsf{f} \leq \varsigma(\lambda, n)$. Here, ς is a fixed negligible function.

Both the OPS one-time private Hamming LDC and our constructed one-time private InsDel LDC are secure against information theoretic adversaries, so long as the secret key is picked uniformly at random. However, it is possible to also pick the secret key in a *psuedo-random* manner and obtain security against any class of PPT adversaries, assuming the existence of one-way functions.

Ovstrovsky, Pandey, and Sahai also give a construction of a private locally decodable code that is secure even when the adversary is given access to *polynomially-many* (in the security parameter) codewords (i.e., it is not one-time). The construction relies on a family of *psuedo-random functions* and is therefore secure against any class of PPT adversaries, assuming the existence of one-way functions. We emphasize that applying our compiler on this "multi-time" private Hamming code yields a secure "multi-time" private InsDel code.

Definition 5: Let λ be the security parameter. A code $C[K, k, q_1, q_2, \lambda]$ consisting of a tuple of PPT algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is a $(\ell, \rho, p, \text{dist})$-*private locally decodable code* if:

- $\mathsf{Gen}(1^\lambda)$ is the key generation algorithm that takes 1^λ as input and outputs a secret key $\mathsf{sk} \in \{0, 1\}^*$, for security parameter λ;
- $\mathsf{Enc}: \Sigma_1^k \times \{0, 1\}^* \to \Sigma_2^K$ is the encoding algorithm that takes as input a message $x \in \Sigma_1^k$ and a secret key sk and outputs a codeword $y \in \Sigma_2^K$; and
- $\mathsf{Dec}^{y'}: \{0, 1\}^{\log k} \times \{0, 1\}^* \to \Sigma_1$ is the decoding algorithm that takes as input index $i \in [k]$ and secret key sk, and is additionally given query access to a corrupted codeword $y' \in \Sigma_2^{K'}$ and outputs $b \in \Sigma_1$ after making at most ℓ queries to y'.

- Let Fool be a predicate such that $\mathsf{Fool}(y', \rho, p, \mathsf{sk}, x, y) = 1$ if and only if 1) $\text{dist}(y, y') \leq \rho$; and 2) $\exists i \in [k]$ such that $\Pr[\mathsf{Dec}^{y'}(i, \mathsf{sk}) = x_i] < p$, where the probability is taken over the random coins of Dec; and $\mathsf{Fool}(y', \rho, p, \mathsf{sk}, x, y) = 0$ otherwise. Consider the $\texttt{priv-LDC-Game}$ defined in Fig. 1. We require that for all PPT adversaries $\mathcal{A}$ there exists a negligible function $\varepsilon(\cdot)$ such that

$$\Pr[\texttt{priv-LDC-Game}(\mathcal{A}, C, 1^\lambda, \mathsf{Fool}) = 1] \leq \varepsilon(\lambda).$$

$\texttt{priv-LDC-Game}(\mathcal{A}, C, 1^\lambda, \mathsf{Fool})$:
Input: A PPT adversary $\mathcal{A}$, a locally decodable code $C[K, k, q_1, q_2, \lambda]$ with PPT algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, security parameter 1^λ, and predicate Fool.
1) Obtain $\mathsf{sk} \leftarrow \mathsf{Gen}(1^\lambda)$ and share sk with Enc and Dec. Note $\mathcal{A}$ is not given sk.
2) For $i \in [h]$, where $h = \text{poly}(k)$ is an integer:
   a) $x_i \leftarrow \mathcal{A}(1^\lambda, (x_1, y_1), \ldots (x_{i-1}, y_{i-1}))$.
   b) $y_i \leftarrow \mathsf{Enc}(x_i; \mathsf{sk})$;
   c) $y_i' \leftarrow \mathcal{A}(1^\lambda, (x_1, y_1), \ldots (x_i, y_i))$.
The output of $\texttt{priv-LDC-Game}$ is 1 if there exists $i \in [h]$ such that $\mathsf{Fool}(y_i', \rho, p, \mathsf{sk}, x_i, y_i) = 1$; else $\texttt{priv-LDC-Game}$ outputs 0.

Fig. 1. Definition of $\texttt{priv-LDC-Game}$.

Remark 2: Definition 5 differs slightly the original definition proposed in Ovstrovsky, Pandey, and Sahai [OPS07] in that we allow the attacker to output a corrupted codeword y_i' in every round $i \leq h$, while in [OPS07] the attacker only attempts to corrupt the codeword in the last round. However, the two definitions are equivalent, up to a $1/\text{poly}(\lambda)$ loss in the security. In particular, an attacker that breaks Definition 5 can efficiently be transformed into an attacker that breaks the definition from [OPS07] i.e., we simply guess the index $i' \leq h$ of the first round in which the attacker is successful.

For security parameter λ, the "multi-time" private Hamming code of [OPS07] has constant rate, locality $\omega(\log^2(\lambda))$, constant error-rate, and success probability $1 - \mathsf{negl}(\lambda)$, for some negligible function.

Corollary 2: Assume that one-way functions exist and let $\ell_\mathsf{f} := \ell_\mathsf{f}(\lambda, n) = \omega(\log^2(\lambda) \cdot O(\log^4(n)))$. There exists a binary code $C_\mathsf{f}[n, k, q_1, 2, \lambda]$ that is a $(\ell_\mathsf{f}, \rho_\mathsf{f}, p_\mathsf{f})$-private InsDel LDC (as per Definition 5) with constant information rate $k/n = \Theta(1)$, where $\rho_\mathsf{f} = \Theta(1)$, and $p_\mathsf{f} = \Theta(1)$.

Remark 3: The reduction for Corollary 2 is nearly identical to that of Theorem 1, except we must now account for $\text{poly}(k)$ rounds where the adversary attempts to fool the decoder. An identical argument uses Lemma 1 to that the probability of succeeding in each individual round of $\texttt{priv-LDC-Game}$ is negligible. Since there are only polynomially many rounds the the probability that the attacker succeeds in any of the rounds is still negligible.

B. Resource-Bounded InsDel Locally Decodable Code Construction

Next we use Theorem 2 with the resource-bounded Hamming LDC of Blocki, Kulkarni, and Zhou [BKZ20] which works for *any* class $\mathbb{C}$ that admits a *safe function*. A function $f\colon \{0,1\}^n \to \{0,1\}^*$ is δ-*safe* for a class $\mathbb{C}$ of algorithms if for all $\mathcal{A} \in \mathbb{C}$ we have $\Pr[\mathcal{A}(x) = f(x)] \leq \delta$, where the probability is taken over the random coins of $\mathcal{A}$ and the selection of an input $x \in \{0,1\}^n$. The code construction of [BKZ20] is in the (parallel) *random oracle model*, where the encoder and decoder additionally have access to some random oracle H. For security parameter λ and fixed negligible functions ϑ_1, ϑ_2, their code has constant-rate, locality $\mathrm{polylog}(\lambda)$, constant error-rate, success probability $1 - \vartheta_1(\lambda)$, and security $\varepsilon \leq \vartheta_2(\lambda) + q\delta$, where q is an upper bound on the number of oracle queries made by any algorithm in $\mathbb{C}$.

In the (parallel) random oracle model one can provably establish the existence of safe functions for many natural classes of channels; e.g., space bounded or sequential time bounded. As an example, if $\mathsf{H}\colon \{0,1\}^* \to \{0,1\}^\lambda$ is a random oracle then the function $\mathsf{H}^{T+1}(\cdot)$ is $\delta = q \cdot T \cdot 2^{-\lambda}$-safe against the class of algorithms making at most q total queries to H over *at most T* rounds. Similar results holds for the classes of space-bounded or space-time bounded channels. The class of sequentially bounded channels is motivated by the observation that the depth of computation that the channel performs is restricted in most natural settings; e.g., traveling at the speed of light, it would take between 3 and 22.3 minutes for a transmission from Mars to reach Earth (the exact time would depend on the current orbital location of the planets).

Corollary 3: Let λ be a security parameter, let $\mathbb{C}$ be a class of algorithms in the parallel random oracle model admitting a δ-safe function, and let $k = \mathrm{poly}(\lambda)$. For random oracle H there exists a binary code $C_{\mathsf{f}}^{\mathsf{H}}[n,k,2]$ that is a $(\ell_{\mathsf{f}}, \rho_{\mathsf{f}}, p_{\mathsf{f}}, \varepsilon_{\mathsf{f}}, \mathbb{C})$-InsDel LDC against class $\mathbb{C}$, where $\ell_{\mathsf{f}} = \mathrm{polylog}(\lambda) \cdot \log^4(n')$, $\rho_{\mathsf{f}} = \Theta(1)$, $p_{\mathsf{f}} = \Theta(1)$, and $\varepsilon_{\mathsf{f}} \leq \varsigma(\lambda, n') - q \cdot \delta$. Here, q is an upper bound on the total queries any algorithm in $\mathbb{C}$ makes to H, ς is a fixed negligible function, and n' is the length of a word received by the decoder.

Remark 4: While the construction of [BKZ20] relies on the random oracle model we stress that this dependence is not inherent to our results. Given any standard model construction of a Hamming LDC for resource bounded channels we could similarly obtain a standard model InsDel LDC for resource bounded channels by applying Theorem 2. Thus, it is plausible that one could replace the random oracle model assumption with, for example, the assumption that time-lock puzzles [RSW96, BN00, GMPY11, MMV11, BGJ+16] exist.

In particular, Blocki, Kulkarni, and Zhou [BKZ20] provide serveral examples of safe functions in various models to construct resource-bounded Hamming LDCs. These include safe functions secure in the parallel random oracle model, safe functions which are secure against sequential time-bounded adversaries, and safe functions based on graphs with sufficiently large pebbling costs.

REFERENCES

[BBG+20] Alexander R. Block, Jeremiah Blocki, Elena Grigorescu, Shubhang Kulkarni, and Minshen Zhu. Locally decodable/correctable codes for insertions and deletions. In Nitin Saxena and Sunil Simon, editors, *40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 182 of *LIPIcs*, pages 16:1–16:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.FSTTCS.2020.16.

[BGGZ19] J. Blocki, V. Gandikota, E. Grigorescu, and S. Zhou. Relaxed locally correctable codes in computationally bounded channels*. In *2019 IEEE International Symposium on Information Theory (ISIT)*, pages 2414–2418, 2019. doi:10.1109/ISIT.2019.8849322.

[BGJ+16] Nir Bitansky, Shafi Goldwasser, Abhishek Jain, Omer Paneth, Vinod Vaikuntanathan, and Brent Waters. Time-lock puzzles from randomized encodings. In Madhu Sudan, editor, *ITCS 2016*, pages 345–356. ACM, January 2016. doi:10.1145/2840728.2840745.

[BGZ18] J. Brakensiek, V. Guruswami, and S. Zbarsky. Efficient low-redundancy codes for correcting multiple deletions. *IEEE Transactions on Information Theory*, 64(5):3403–3410, 2018. doi:10.1109/TIT.2017.2746566.

[BKZ20] Jeremiah Blocki, Shubhang Kulkarni, and Samson Zhou. On locally decodable codes in resource bounded channels. In Yael Tauman Kalai, Adam D. Smith, and Daniel Wichs, editors, *ITC 2020*, pages 16:1–16:23. Schloss Dagstuhl, June 2020. doi:10.4230/LIPIcs.ITC.2020.16.

[BN00] Dan Boneh and Moni Naor. Timed commitments. In Mihir Bellare, editor, *CRYPTO 2000*, volume 1880 of *LNCS*, pages 236–254. Springer, Heidelberg, August 2000. doi:10.1007/3-540-44598-6_15.

[CGHL20] Kuan Cheng, Venkatesan Guruswami, Bernhard Haeupler, and Xin Li. Efficient linear and affine codes for correcting insertions/deletions, 2020. arXiv:2007.09075.

[CHL+19] Kuan Cheng, Bernhard Haeupler, Xin Li, Amirbehshad Shahrasbi, and Ke Wu. Synchronization strings: Highly efficient deterministic constructions over small alphabets. In Timothy M. Chan, editor, *30th SODA*, pages 2185–2204. ACM-SIAM, January 2019. doi:10.1137/1.9781611975482.132.

[CJLW18] Kuan Cheng, Zhengzhong Jin, Xin Li, and Ke Wu. Deterministic document exchange protocols, and almost optimal binary codes for edit errors. In Mikkel Thorup, editor, *59th FOCS*, pages 200–211. IEEE Computer Society Press, October 2018. doi:10.1109/FOCS.2018.00028.

[CJLW19] Kuan Cheng, Zhengzhong Jin, Xin Li, and Ke Wu. Block edit errors with transpositions: Deterministic document exchange protocols and almost optimal binary codes. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *ICALP 2019*, volume 132 of *LIPIcs*, pages 37:1–37:15. Schloss Dagstuhl, July 2019. doi:10.4230/LIPIcs.ICALP.2019.37.

[CL20] Kuan Cheng and Xin Li. Efficient document exchange and error correcting codes with asymmetric information, 2020. arXiv:2007.00870.

[CLZ20] Kuan Cheng, Xin Li, and Yu Zheng. Locally decodable codes with randomized encoding. Cryptology ePrint Archive, Report 2020/031, 2020. https://eprint.iacr.org/2020/031.

[DGY10] Zeev Dvir, Parikshit Gopalan, and Sergey Yekhanin. Matching vector codes. In *51st FOCS*, pages 705–714. IEEE Computer Society Press, October 2010. doi:10.1109/FOCS.2010.73.

[Efr09] Klim Efremenko. 3-query locally decodable codes of subexponential length. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 39–44. ACM Press, May / June 2009. doi:10.1145/1536414.1536422.

[GHS20] Venkatesan Guruswami, Bernhard Haeupler, and Amirbehshad Shahrasbi. Optimally resilient codes for list-decoding from insertions and deletions. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *52nd ACM STOC*, pages 524–537. ACM Press, June 2020. doi:10.1145/3357713.3384262.

[GL18] Venkatesan Guruswami and Ray Li. Coding against deletions in oblivious and online models. In Artur Czumaj, editor, *29th SODA*, pages 625–643. ACM-SIAM, January 2018. doi:10.1137/1.9781611975031.41.

[GL19] V. Guruswami and R. Li. Polynomial time decodable codes for the binary deletion channel. *IEEE Transactions on Information Theory*, 65(4):2171–2178, 2019. doi:10.1109/TIT.2018.2876861.

[GMPY11] Juan A. Garay, Philip D. MacKenzie, Manoj Prabhakaran, and Ke Yang. Resource fairness and composability of cryptographic protocols. *Journal of Cryptology*, 24(4):615–658, October 2011. doi:10.1007/s00145-010-9080-z.

[GS16] Venkatesan Guruswami and Adam Smith. Optimal rate code constructions for computationally simple channels. *J. ACM*, 63(4), September 2016. doi:10.1145/2936015.

[GW17] V. Guruswami and C. Wang. Deletion codes in the high-noise and high-rate regimes. *IEEE Transactions on Information Theory*, 63(4):1961–1970, 2017. doi:10.1109/TIT.2017.2659765.

[Hae19] Bernhard Haeupler. Optimal document exchange and new codes for insertions and deletions. In David Zuckerman, editor, *60th FOCS*, pages 334–347. IEEE Computer Society Press, November 2019. doi:10.1109/FOCS.2019.00029.

[HRS19] Bernhard Haeupler, Aviad Rubinstein, and Amirbehshad Shahrasbi. Near-linear time insertion-deletion codes and $(1+\epsilon)$-approximating edit distance via indexing. In Moses Charikar and Edith Cohen, editors, *51st ACM STOC*, pages 697–708. ACM Press, June 2019. doi:10.1145/3313276.3316371.

[HS17] Bernhard Haeupler and Amirbehshad Shahrasbi. Synchronization strings: codes for insertions and deletions approaching the singleton bound. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *49th ACM STOC*, pages 33–46. ACM Press, June 2017. doi:10.1145/3055399.3055498.

[HS18] Bernhard Haeupler and Amirbehshad Shahrasbi. Synchronization strings: explicit constructions, local decoding, and applications. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *50th ACM STOC*, pages 841–854. ACM Press, June 2018. doi:10.1145/3188745.3188940.

[HSS18] Bernhard Haeupler, Amirbehshad Shahrasbi, and Madhu Sudan. Synchronization strings: List decoding for insertions and deletions. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *ICALP 2018*, volume 107 of *LIPIcs*, pages 76:1–76:14. Schloss Dagstuhl, July 2018. doi:10.4230/LIPIcs.ICALP.2018.76.

[KLM04] Marcos Kiwi, Martin Loebl, and Jiří Matoušek. Expected length of the longest common subsequence for large alphabets. In Martín Farach-Colton, editor, *LATIN 2004: Theoretical Informatics*, pages 302–311, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[KMRZS17] Swastik Kopparty, Or Meir, Noga Ron-Zewi, and Shubhangi Saraf. High-rate locally correctable and locally testable codes with sub-polynomial query complexity. *J. ACM*, 64(2), May 2017. doi:10.1145/3051093.

[KS16] Swastik Kopparty and Shubhangi Saraf. Guest column: Local testing and decoding of high-rate error-correcting codes. *SIGACT News*, 47(3):46–66, August 2016. doi:10.1145/2993749.2993761.

[KT00] Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *32nd ACM STOC*, pages 80–86. ACM Press, May 2000. doi:10.1145/335305.335315.

[KW03] Iordanis Kerenidis and Ronald de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. In *35th ACM STOC*, pages 106–115. ACM Press, June 2003. doi:10.1145/780542.780560.

[Lev66] Vladimir Iosifovich Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966. Doklady Akademii Nauk SSSR, V163 No4 845-848 1965.

[Lip94] Richard J. Lipton. A new approach to information theory. In Patrice Enjalbert, Ernst W. Mayr, and Klaus W. Wagner, editors, *STACS 94*, pages 699–708, Berlin, Heidelberg, 1994. Springer Berlin Heidelberg.

[LTX20] Shu Liu, Ivan Tjuawinata, and Chaoping Xing. On list decoding of insertion and deletion errors, 2020. arXiv:1906.09705.

[MBT10] H. Mercier, V. K. Bhargava, and V. Tarokh. A survey of error-correcting codes for channels with symbol synchronization errors. *IEEE Communications Surveys Tutorials*, 12(1):87–96, 2010. doi:10.1109/SURV.2010.020110.00079.

[Mit08] Michael Mitzenmacher. A survey of results for deletion channels and related synchronization channels. In Joachim Gudmundsson, editor, *Algorithm Theory – SWAT 2008*, pages 1–3, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[MMV11] Mohammad Mahmoody, Tal Moran, and Salil P. Vadhan. Time-lock puzzles in the random oracle model. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 39–50. Springer, Heidelberg, August 2011. doi:10.1007/978-3-642-22792-9_3.

[MPSW05] Silvio Micali, Chris Peikert, Madhu Sudan, and David A. Wilson. Optimal error correction against computationally bounded noise. In Joe Kilian, editor, *TCC 2005*, volume 3378 of *LNCS*, pages 1–16. Springer, Heidelberg, February 2005. doi:10.1007/978-3-540-30576-7_1.

[OPC15] Rafail Ostrovsky and Anat Paskin-Cherniavsky. Locally decodable codes for edit distance. In Anja Lehmann and Stefan Wolf, editors, *Information Theoretic Security*, pages 236–249, Cham, 2015. Springer International Publishing.

[OPS07] Rafail Ostrovsky, Omkant Pandey, and Amit Sahai. Private locally decodable codes. In Lars Arge, Christian Cachin, Tomasz Jurdzinski, and Andrzej Tarlecki, editors, *ICALP 2007*, volume 4596 of *LNCS*, pages 387–398. Springer, Heidelberg, July 2007. doi:10.1007/978-3-540-73420-8_35.

[RSW96] Ronald L Rivest, Adi Shamir, and David A Wagner. Time-lock puzzles and timed-release crypto. 1996.

[SB19] Jin Sima and Jehoshua Bruck. Optimal k-deletion correcting codes. In *2019 IEEE International Symposium on Information Theory (ISIT)*, pages 847–851, 2019. doi:10.1109/ISIT.2019.8849750.

[SGB20a] Jin Sima, Ryan Gabrys, and Jehoshua Bruck. Optimal codes for the q-ary deletion channel. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pages 740–745, 2020. doi:10.1109/ISIT44484.2020.9174241.

[SGB20b] Jin Sima, Ryan Gabrys, and Jehoshua Bruck. Optimal systematic t-deletion correcting codes. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pages 769–774, 2020. doi:10.1109/ISIT44484.2020.9173986.

[Slo02] N.J.A. Sloane. On single-deletion-correcting codes. *arXiv: Combinatorics*, 2002.

[SS16] Ronen Shaltiel and Jad Silbak. Explicit List-Decodable Codes with Optimal Rate for Computationally Bounded Channels. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2016)*, pages 45:1–45:38, 2016. URL: http://drops.dagstuhl.de/opus/volltexte/2016/6668, doi:10.4230/LIPIcs.APPROX-RANDOM.2016.45.

[STV99] M. Sudan, L. Trevisan, and S. Vadhan. Pseudorandom generators without the xor lemma. In *Proceedings. Fourteenth Annual IEEE Conference on Computational Complexity (Formerly: Structure in Complexity Theory Conference) (Cat.No.99CB36317)*, pages 4–, 1999. doi:10.1109/CCC.1999.766253.

[SZ99] Leonard J. Schulman and David Zuckerman. Asymptotically good codes correcting insertions, deletions, and transpositions. *IEEE Trans. Inf. Theory*, 45(7):2552–2557, 1999. doi:10.1109/18.796406.

[Yek08] Sergey Yekhanin. Towards 3-query locally decodable codes of subexponential length. *J. ACM*, 55(1), February 2008. doi:10.1145/1326554.1326555.

[Yek12] Sergey Yekhanin. Locally decodable codes. *Foundations and Trends® in Theoretical Computer Science*, 6(3):139–255, 2012. URL: http://dx.doi.org/10.1561/0400000030, doi:10.1561/0400000030.