Lossy Cryptography from Code-Based Assumptions

Dense-Sparse LPN: A New Subexponentially Hard LPN Variant in SZK

Quang Dao* Aayush Jain†

January 6, 2025

Abstract

Over the past few decades, we have seen a proliferation of advanced cryptographic primitives with lossy or homomorphic properties built from various assumptions such as Quadratic Residuosity, Decisional Diffie-Hellman, and Learning with Errors. These primitives imply hard problems in the complexity class $\mathcal{SZK}$ (statistical zero-knowledge); as a consequence, they can only be based on assumptions that are broken in $\mathcal{BPP}^{\mathcal{SZK}}$. This poses a barrier for building advanced cryptography from code-based assumptions such as Learning Parity with Noise (LPN), as LPN is only known to be in $\mathcal{BPP}^{\mathcal{SZK}}$ under an extremely low noise rate $\frac{\log^2 n}{n}$, for which it is broken in quasi-polynomial time.

In this work, we propose a new code-based assumption: Dense-Sparse LPN, that falls in the complexity class $\mathcal{BPP}^{\mathcal{SZK}}$ and is conjectured to be secure against subexponential time adversaries. Our assumption is a variant of LPN that is inspired by McEliece's cryptosystem and random k-XOR in average-case complexity. Roughly, the assumption states that

$$(\mathbf{T}\,\mathbf{M}, \mathbf{s}\,\mathbf{T}\,\mathbf{M} + \mathbf{e}) \quad \text{is indistinguishable from} \quad (\mathbf{T}\,\mathbf{M}, \mathbf{u}),$$

for a random (dense) matrix $\mathbf{T}$, random sparse matrix $\mathbf{M}$, and sparse noise vector $\mathbf{e}$ drawn from the Bernoulli distribution with inverse polynomial noise probability.

We leverage our assumption to build lossy trapdoor functions (Peikert-Waters STOC 08). This gives the first post-quantum alternative to the lattice-based construction in the original paper. Lossy trapdoor functions, being a fundamental cryptographic tool, are known to enable a broad spectrum of both lossy and non-lossy cryptographic primitives; our construction thus implies these primitives in a generic manner. In particular, we achieve collision-resistant hash functions with plausible subexponential security, improving over a prior construction from LPN with noise rate $\frac{\log^2 n}{n}$ that is only quasi-polynomially secure.

*Carnegie Mellon University. Email: qvd@andrew.cmu.edu
†Carnegie Mellon University. Email: aayushja@andrew.cmu.edu

Contents

1 Introduction

Introduced in 2005, the Learning with Errors (LWE) assumption [Reg05] has emerged as a basis for designing post-quantum cryptography. LWE and its structured variants such as Ring-LWE [LPR10] or NTRU [HPS98] are central to constructing a host of advanced cryptographic primitives including fully homomorphic encryption for classical [Gen09, BV11, GSW13] and quantum computations [Mah18a, Bra18], attribute-based and other advanced encryption schemes [GVW13, GVW15], non-interactive zero-knowledge [PS19], succinct arguments [CJJ22], and many other advances in classical [GKW17, WZ17, GKW18, LMW23] and quantum cryptography [BCM+18, Mah18b].

While LWE has proven to be surprisingly expressive in giving rise to advanced primitives, other post-quantum assumptions such as Learning Parity with Noise [BFKL94], Isogenies [Cou06, RS06, CLM+18], and Multivariate Quadratic [OSS84], currently stand nowhere close in implying such advanced primitives, making LWE the single point of failure for designing advanced post-quantum cryptography. This state of affairs is highly unsatisfactory, since we would like to have some diversity in the assumptions implying a given primitive to hedge against unexpected cryptanalytic breakthroughs. Indeed, recent works [CD23a, MMP+23, Rob23] have rendered the once-believed post-quantum assumption of SIDH classically broken in polynomial time.

This work aims to address a *potential stagnation* in terms of techniques and assumptions implying advanced post-quantum cryptography. This lack of versatility in assumptions for the most part can be attributed to the *lack of techniques* in utilizing other post-quantum assumptions. The focus of this work lies in code-based cryptographic assumptions such as the Learning Parity with Noise (LPN) assumption [BFKL94] and its variants.

Learning Parity with Noise posits that random linear equations (with a planted secret solution) that is perturbed by sparse noise appears pseudorandom. Namely:

$$(\mathbf{A}, \mathbf{s} \cdot \mathbf{A} + \mathbf{e}) \quad \text{is indistinguishable from} \quad (\mathbf{A}, \mathbf{u}),$$

where the coefficient matrix $\mathbf{A}$ is chosen at random from $\mathbb{F}_2^{n \times m}$, the secret $\mathbf{s} \leftarrow \mathbb{F}_2^{1 \times n}$, $\mathbf{u}$ is chosen to be a random vector in $\mathbb{F}_2^{1 \times m}$ and the error vector $\mathbf{e} \in \mathbb{F}_2^{1 \times m}$ is chosen so that each coordinate is i.i.d. sampled from the Bernoulli distribution with probability ϵ. The problem is believed to be *subexponentially secure*, meaning that subexponential $\exp(n^{O(1)})$-time adversaries have negligible distinguishing advantage when $\epsilon = O\left(\frac{1}{n^\delta}\right)$ for any constant $\delta \in (0, 1)$.[1]

LPN is conceptually similar to LWE, in the sense that both posit pseudorandomness of planted random linear equations perturbed with noise. However, while for LWE the noise has low magnitude, for LPN it is sparse. One would expect that due to this similarity, LPN should imply a comparable variety of advanced primitives—yet this could not be any further from reality. On the one hand, recent works have leveraged LPN (over large fields) in combination with Bilinear Maps [MVO91] and Goldreich's PRG [Gol11] to build indistinguishability obfuscation [JLS21, JLS22].

On the other hand, despite almost three decades of research and in drastic contrast to LWE, we currently know only a handful of ways to leverage the LPN assumption. This is evident in the fact that aside from CPA/CCA secure public-key encryption schemes [Ale03, DMN12, KMP14, YZ16] and UC-secure two-round oblivious transfer [DGH+20], subexponentially secure variants of LPN alone are currently not known to imply any other primitives in Cryptomania [Imp95]. Things seem to improve when one works with the quasi-polynomial time broken variant of the LPN

[1] A stronger version of subexponential security, which we do not consider in this work, also requires that the distinguishing advantage is an inverse subexponential $\exp(-n^{O(1)})$.

assumption with very small noise probability $O(\frac{\log^2 n}{n})$, but even assuming this variant, very few primitives are known. These include collision-resistant and collapsing hash functions [BLVW19, YZW+19, Zha22], identity-based encryption [BLSV18], and statistically-sender-private oblivious transfer [BF22].

This brings us to our goal:

Goal. *Devise new coding-theoretic techniques and assumptions for building advanced cryptography.*

To facilitate progress on the main goal above, we focus more on identifying properties of the assumption that could enable progress on the question, rather than focusing on specific primitives themselves. What makes assumptions such as LWE, Diffie-Hellman, Bilinear Maps, or Quadratic Reciprocity, special is that they can be used to design primitives with "lossy" or "homomorphic" properties, such as lossy trapdoor functions [PW08] and linearly homomorphic encryption. Furthermore, the homomorphic/lossy properties of the assumption make them easier to work with to design other advanced Cryptomania primitives, such as attribute-based encryption or succinct arguments.

A key property that captures such assumptions is that they can be broken using an $\mathcal{SZK}$ oracle, where $\mathcal{SZK}$ is the complexity class of languages that have statistically-hiding zero-knowledge proofs. This "$\mathcal{SZK}$-broken" complexity class, known as $\mathcal{BPP}^{\mathcal{SZK}}$, consists of languages that can be decided efficiently using a *statistical difference* (SD) oracle [SV97]. The SD oracle takes as input two polynomial sized distribution samplers $(\mathcal{D}_0, \mathcal{D}_1)$ (represented as randomized circuits), with the promise that either the statistical distance between the distributions is less than $\frac{1}{3}$ or it is more than $\frac{2}{3}$. The oracle then identifies which is the case.

This $\mathcal{SZK}$ regime indeed captures all of the assumptions mentioned above. For LPN, it is known [BLVW19] that the quasi-polynomial time broken variant with noise probability $O(\frac{\log^2 n}{n})$ can be broken with an $\mathcal{SZK}$ oracle, whereas subexponential time secure variants with inverse polynomial noise probability $\frac{1}{n^\delta}$ for $\delta \in (0,1)$ are currently not known to be in $\mathcal{BPP}^{\mathcal{SZK}}$.[2] This helps to explain why so little is known from LPN with inverse polynomial noise rate.

Therefore, to make progress in code-based cryptography, the first step would be to answer the following question:

Question. *Is there a subexponentially-secure variant of LPN with inverse polynomial noise probability that is in $\mathcal{BPP}^{\mathcal{SZK}}$?*

In this work, we are mainly concerned with variants of LPN, which are assumptions of the form $(\mathbf{A}, \mathbf{sA} + \mathbf{e}) \approx_c (\mathbf{A}, \mathbf{u})$ for a possibly structured matrix $\mathbf{A}$. This is because such LPN-type assumption seemingly contains enough structure to build interesting primitives. If we relax the requirement to *any* coding-theoretic assumption, then indeed we can find assumptions such as *code equivalence* that are also in $\mathcal{BPP}^{\mathcal{SZK}}$ [PR97], yet by itself do not appear to give us any advanced cryptography (besides signatures in the random oracle model [Gir90, BMPS20]).

1.1 Our Results

We introduce a novel, well-motivated variant of LPN that we believe is secure against subexponential time algorithms. Unlike LPN where the matrix is chosen randomly, in our case it is a structured matrix. We work with a sufficiently small but inverse polynomial probability $\frac{1}{n^\delta}$ for a constant $\delta \in (0.5, 1)$, a regime for which our assumption can be conjectured to be hard against

[2] In this work, we say that an assumption is in a complexity class if it can be broken by an adversary in that complexity class.

subexponential time adversaries.[3] Since our assumption implies primitives that can be broken by $\mathcal{BPP}^{\mathcal{SZK}}$,[4] our assumption indeed lies in $\mathcal{BPP}^{\mathcal{SZK}}$, making it the first plausible subexponential-time secure LPN variant known to be in $\mathcal{BPP}^{\mathcal{SZK}}$.

New Assumption: Dense-Sparse LPN. Our assumption borrows structural properties of two well-studied assumptions: the standard Learning Parity with Noise [BFKL94] and the sparse Learning Parity with Noise problem [Ale03]. Introduced in 2003 by Alekhnovich, Sparse LPN is exactly like standard LPN except that each column is chosen randomly among k-sparse vectors, where $k \geq 3$ is a constant. Sparse LPN is closely related to well-studied problems in the domain of constraint satisfaction and local pseudorandom generators [Gol00, CM01, Fei02, MST03, FKO06, CEMT09, BQ09, ABW10, ABR12, BQ12, App12, App13, OW14, AL16, KMOW17, CDM+18, AK19], and when the number of samples satisfies $m = n^{\frac{k}{2}(1-\rho)}$ for any constant $0 < \rho < 1$,[5] it is believed to be subexponentially secure (provided the noise probability is a large enough inverse polynomial). Sparse LPN has also been shown to give rise to public-key encryption by Applebaum, Barak and Wigderson [ABW10], but not more advanced Cryptomania primitives.

Our assumption combines features from both LPN and Sparse LPN, and posits that LPN holds for the following *Dense-Sparse* matrix distribution. We first sample a k-sparse matrix $\mathbf{M} \in \mathbb{F}_2^{n \times m}$ according to the distribution of coefficient matrix for the Sparse LPN assumption. We then sample a random (dense) matrix $\mathbf{T} \leftarrow \mathbb{F}_2^{n' \times n}$, where $n' = \alpha n$ for some constant $\alpha \in (0, 1)$ (for simplicity, we set $\alpha = 1/2$ in our paper). Finally, we give out $\mathbf{A} = \mathbf{T} \cdot \mathbf{M} \in \mathbb{F}_2^{n/2 \times m}$, and assume that random codewords of $\mathbf{A}$, perturbed by a Bernoulli noise vector $\mathbf{e}$ of inverse-polynomial noise rate, look pseudorandom. More formally, our assumption is stated as follows.

Assumption 1.1 (Dense-Sparse LPN, informal)**.** *Let $k \geq 3$ be a constant, and consider parameters $n \in \mathbb{N}$, $m = m(n) < n^{k/2}$, and $\epsilon = \epsilon(n) < 1$. Let $\mathcal{M}_{\mathsf{sp}}$ be an efficiently sampleable "good" distribution over all k-sparse matrices in $\mathbb{F}_2^{n \times m}$. We say that the $(n, m, k, \mathcal{M}_{\mathsf{sp}}, \epsilon)$-*Dense-Sparse LPN *assumption holds if the following two distributions are computationally indistinguishable:*

$$\{(\mathbf{T}\,\mathbf{M}, \mathbf{s}\,\mathbf{T}\,\mathbf{M} + \mathbf{e})\}_{n \in \mathbb{N}} \approx_c \{(\mathbf{T}\,\mathbf{M}, \mathbf{u})\}_{n \in \mathbb{N}},$$

where $\mathbf{T} \leftarrow \mathbb{F}_2^{n/2 \times n}$, $\mathbf{M} \leftarrow \mathcal{M}_{\mathsf{sp}}$, $\mathbf{s} \leftarrow \mathbb{F}_2^{1 \times n/2}$, $\mathbf{e} \leftarrow \mathsf{Ber}(\epsilon)^{1 \times m}$, and $\mathbf{u} \leftarrow \mathbb{F}_2^m$.

Looking ahead, our constructions will require us to assume Dense-Sparse LPN for an inverse polynomial noise rate $\epsilon = O(n^{-\delta})$ for some constant δ close to 1, and the number of samples $m = \Omega(n^{1+\rho(\delta)})$ for a constant ρ that depends on δ. This parameter regime is plausibly secure against subexponential-time adversaries. In particular, we believe that as long as the number of samples $m < n^{1+(k/2-1)(1-\rho)}$ for some constant $k \geq 3$ and $\rho \in (0, 1)$, the assumption should be secure against adversaries that run in time that is smaller than $\min\left(2^{\widetilde{O}(n^{\rho})}, 2^{\widetilde{O}(n \cdot \epsilon)}\right)$. We refer to Section 7 for details on our cryptanalysis, and also to Table 1 for conjectured security of Dense-Sparse LPN in the context of cryptographic applications.

An important technical point in Assumption 1.1 is that of a "good" distribution of k-sparse matrices. This is due to the following reason: for $\mathbf{M} \in \mathbb{F}_2^{n \times m}$ chosen uniformly at random from

[3] However, an adversary's success probability in breaking our assumption is at least inverse quasi-polynomial; see Section 2.3 for more discussion.

[4] It's a folklore result that lossy trapdoor functions, which we construct, lie in $\mathcal{BPP}^{\mathcal{SZK}}$; a formal proof can be found in [FR23, Appendix B].

[5] When $m = \Omega(n^{k/2})$, due to the birthday bound two equations will repeat with constant probability, implying a trivial cheating strategy.

the set of all k-sparse matrices, there is an *inverse polynomial* probability of $O(m^2/n^k)$ that $\mathbf{M}$ has a vector $\mathbf{x}$ in its kernel of constant Hamming weight (so that $\mathbf{M}\,\mathbf{x} = \mathbf{0}$). When this "bad" event happens, one cannot hope for distinguishing security to hold. Thus, since we want our distinguishing advantage to be negligible, we must sample $\mathbf{M}$ from another "good" distribution where this "bad" event happens with negligible probability; in particular, we will use the recent distribution constructed by Applebaum and Kachlon [AK19]. We will expand on this in Section 2.3.

Connections to McEliece. Our assumption can be viewed as applying the design principles of the classic McEliece [McE78] and Niederreiter [Nie86] cryptosystems, which is to hide the sparse matrix $\mathbf{M}$ whose exposure would lead to an efficient attack in our parameter regime. In this sense, we follow a rich body of works on McEliece instantiated with different families of codes [Sid94, LJ12, BL05, BLP10, BLP11, JM96, MTSB12, SK14, HSEA14]. Nevertheless, there are two important distinctions between our assumption and prior McEliece variants. The first is that our variants are *not* algebraically structured, unlike the original McEliece cryptosystem itself (which uses binary Goppa codes), or many other algebraic variants [Sid94, JM96, BL05] or LDPC codes [BC07, BBC08] that have subsequently been broken [SS92, BC07, MS07, OTD10, Wie10, LT13, COT14, BCD+16].[6] Secondly, we diverge from McEliece by making the masking matrix $\boldsymbol{T}$ *compressing* of dimension $\alpha n \times n$ for any $\alpha < 1$, which is necessary for ensuring security in our setting. We expand more on this connection in Section 2.1.

Cryptographic Applications. We leverage Dense-Sparse LPN with inverse polynomial noise rate to build the following two primitives: a *collision-resistant hash function* (in a simple and direct manner), and a *lossy trapdoor function*.

Lossy Trapdoor Functions (LTDFs), introduced by Peikert and Waters in 2008 [PW08], is a fundamental cryptographic tool that has found countless applications to building other cryptographic applications. LTDFs consist of a function family $F_{\mathsf{fk}}(\cdot)$ indexed by a public function key fk, where the algorithm Gen that samples fk could sample keys in two modes. When the mode is injective, then the function $F_{\mathsf{fk}}(\cdot)$ is injective and can even be inverted uniquely using a trapdoor td generated by Gen at the same time of sampling fk. In lossy mode, the range of the function $F_{\mathsf{fk}}(\cdot)$ is significantly smaller than the number of inputs. Equivalently, this also means that the conditional entropy in $x \in \{0,1\}^\ell$, given $y = F_{\mathsf{fk}}(x)$ for a random x is large. In our setting we design such LTDFs for which the conditional entropy is at least $\Omega(\ell)$ where ℓ is the bit length of x. Finally, the two modes are required to be computationally indistinguishable, meaning that it is computationally hard to distinguish a random lossy key from a random injective key.

Theorem 1.1 (informal)**.** *Assuming Dense-Sparse LPN with inverse polynomial noise probability, there exists a construction of LTDF where the lossy mode loses any constant fraction $\Omega(\ell)$ of the input length ℓ.*

We summarize the parameters required for Theorem 1.1 in Table 1; we also give an example of how the parameters can be concretely instantiated. We may set $k = 6$ and $\rho = 1/4$, so that $m = n^{5/2}$ is the number of samples. Then according to Table 1, public-key encryption is possible with error $\epsilon_{\mathsf{PKE}} = \widetilde{O}\left(\frac{1}{n^{1/4}}\right)$,[7] collision-resistant hash functions from error $\epsilon_{\mathsf{CRHF}} = \omega\left(\frac{1}{n^{19/22}}\right)$, and lossy trapdoor functions (as the compression factor $D \to 1$) from $\epsilon_{\mathsf{LTDF}} = \omega\left(\frac{1}{n^{7/10}}\right)$.

[6] In this sense, our assumption is related to the more combinatorial McEliece variant with *medium-density parity check (MDPC)* codes [MTSB12], which still remains secure to this day.

[7] We are using the PKE scheme described in [ABW10] which is designed for Sparse LPN, but it works without change for Dense-Sparse LPN as well.

Application	Error Probability ϵ	Conjectured Security
Public-Key Encryption [ABW10]	$\epsilon_{\mathsf{PKE}} = \frac{1}{n^\rho}$	$\min\left(2^{\widetilde{O}(n^\rho)}, 2^{\widetilde{O}(n^{1-\rho})}\right)$
CRHF (This Work)	$\epsilon_{\mathsf{CRHF}} = \omega\left(\left(\frac{m}{n^{2k}}\right)^{\frac{1}{2k-1}}\right)$	$\min\left(2^{\widetilde{O}(n^\rho)}, 2^{\widetilde{O}(n\cdot\epsilon_{\mathsf{CRHF}})}\right)$
LTDF (This Work)	$\epsilon_{\mathsf{LTDF}} = \omega\left(\left(\frac{m}{n^{Dk}}\right)^{\frac{1}{Dk-1}}\right)$	$\min\left(2^{\widetilde{O}(n^\rho)}, 2^{\widetilde{O}(n\cdot\epsilon_{\mathsf{LTDF}})}\right)$

Table 1: Dense-Sparse LPN parameter regimes and their conjectured security. We work with $m = n^{1+(k/2-1)(1-\rho)}$ samples for any constant $k \geq 3$ and $\rho \in (0,1)$. For a given application in the first column, we give the minimum error probability ϵ (up to polylogarithmic factors) in the second column, and the conjectured security of Dense-Sparse LPN (in terms of adversary runtime) in the third column. For LTDF, we assume a compression factor of $D > 1$, meaning that the lossy mode only retains $(1/D)$-th of the original entropy. CRHF follows from any compression factor $D > 2$.

Lossy Trapdoor Functions are known from a number of quantum-broken assumptions such as Decisional Diffie-Hellman, Bilinear Maps, Quadratic Residuosity, Phi-Hiding, and Decisional Composite Residuosity (DCR). However, prior to our work, no post-quantum assumption barring LWE was known to imply lossy trapdoor functions, including LPN with noise probability $O(\frac{\log^2 n}{n})$ that is broken in quasi-polynomial time.

Since Lossy Trapdoor Functions are known to imply a number of lossy primitives, as a result we can generically realize those primitives from Dense-Sparse LPN. This list of primitives and applications includes: collision-resistant hash functions and CCA secure encryption [PW08], dual-mode commitments and statistically-sender-private oblivious transfer [HLOV11], deterministic encryption [BFO08], trapdoor functions with many hardcore bits, analyzing OAEP [KOS10], hedged public-key encryption with bad randomness [BBN+09], selective opening security [BHY09], pseudo-entropy functions [BHK11], point-function obfuscation [Zha16], computational extractors [DVW20, GKK20], incompressible encodings [MW20], and many more.

1.2 Related Works

LPN Variants and their Applications. Recent works on *pseudorandom correlation generators (PCGs)* [BCGI18, BCG+19, BCG+20b] and *pseudorandom correlation functions (PCFs)* [BCG+20a] have proposed many novel variants of LPN with different matrix distributions [BCG+20a, CRR21, BCG+22, CD23b, RRT23, BCCD23]. While these works are similar to ours in that they introduce new LPN variants, we introduce our variant (Dense-Sparse LPN) for a *different* purpose: building more advanced lossy primitives in Cryptomania. In contrast, it is not known whether PCGs or PCFs imply public-key encryption or other Cryptomania primitives.

Collision-Resistant Hashing from Decoding Problems. Our construction of CRHF builds on the template of multiplying a compressing matrix $\mathbf{A}$ with a "low-norm" vector $\mathbf{x}$. Instantiations of this template from lattices have been around for over two decades [GGH11, Ajt96, LMPR08], and instantiations from codes have also been proposed such as [AFS05, FGS07, BLPS11, AHI+17, BLVW19, YZW+19]. We note that code-based CRHF from this template may base their security on the "binary SVP" (bSVP) assumption instead of LPN, which says that it is difficult to find low-weight vectors in the kernel of a random matrix. However, bSVP by itself does not appear to give

public-key encryption.

Group Actions and SZK Primitives. Besides lattices, certain assumptions on (suitable) group actions [BY91, Cou06] also imply primitives in $\mathcal{SZK}$, and are plausibly post-quantum secure against subexponential time adversaries. In more details, the authors of [ADMP20] showed that group actions satisfying a *weak pseudorandomness* property, such as those based on isogenies like CSIDH [CLM+18] or CSI-FiSh [BKV19], suffices for building a variety of $\mathcal{SZK}$ primitives such as hash proof system [CS02], dual-mode PKE [PVW08], malicious SSP-OT [NP01, HK12], and more. A later work [AMR22] proposed an (almost) 2-to-1 trapdoor claw-free function from group actions, which can be seen as a lossy trapdoor function losing a *single* bit [Mal24]. In contrast, our LTDF construction may lose any constant fraction of all bits, which is strictly stronger.

Future Directions. A fascinating question left open in our work is whether we can construct similar Cryptomania primitives, though not known to generically follow from lossy trapdoor functions, from our Dense-Sparse LPN assumption. These primitives, which are known to be achievable from LPN with quasi-polynomial security, include laconic oblivious transfer [CDG+17, BLSV18], identity-based encryption [BLSV18], and (maliciously-secure) statistically-sender-private oblivious transfer [BF22]. In Section 2.4, we sketch how our current assumption encounters roadblocks toward building these primitives.

2 Overview of Techniques

We now discuss the intuition for how the structural properties of Dense-Sparse LPN puts it in $\mathcal{SZK}$ and enables applications such as lossy trapdoor functions. We then discuss key points in the cryptanalysis of our assumption and end with some open questions.

2.1 Collision-Resistant Hash Functions from Dense-Sparse LPN

Collision-Resistance from LPN. In a nutshell, our progress on building lossy trapdoor functions from Dense-Sparse LPN is a result of achieving *input compression* under a *larger*, inverse polynomial noise rate. In both lattice-based and code-based cryptography, we consider the following hash function family:

$$h_{\mathbf{A}}(\mathbf{x}) = \mathbf{A} \cdot \mathbf{x}, \qquad \text{indexed by } \mathbf{A} \leftarrow \mathcal{R}^{n \times m} \text{ over some finite ring } \mathcal{R},$$

and the input $\mathbf{x}$ comes from a "low-norm" distribution. In the case of LWE, we have $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ for some modulus $q = q(n)$, and $\mathbf{x} \in \{0,1\}^m$. A simple calculation then shows that the number of inputs, which is $|\{0,1\}^m| = 2^m$, is greater than the number of outputs, which is $|\mathbb{Z}_q^n| = 2^{n \log q}$, when $m > n \log q$. This is achievable even for exponential modulus $q = 2^{O(n)}$, for which we only need to set $m > O(n^2) = \mathsf{poly}(n)$.

In contrast, in the case of LPN, we have $\mathbf{A} \leftarrow \mathbb{F}_2^{n \times m}$ and $\mathbf{x} \leftarrow \mathcal{B}(m,t)$, the Hamming ball of weight t in $\mathbb{F}_2^m$. To achieve compression, we then need to ensure that

$$\text{number of inputs for } h_{\mathbf{A}} = \binom{m}{t} \quad \gg \quad 2^n = \text{number of outputs for } h_{\mathbf{A}}. \tag{1}$$

For $m = \mathsf{poly}(n)$, using the binomial approximations $\left(\frac{m}{t}\right)^t \leq \binom{m}{t} \leq \left(\frac{em}{t}\right)^t$, this is only possible when $t = \Omega(n/\log n)$. To see why this is problematic, we now sketch the proof (which can be

found in [BLVW19]) that $h_{\mathbf{A}}$ is collision-resistant, by a reduction to LPN with error probability $\epsilon = \epsilon(n)$. A collision $(\mathbf{x}_0, \mathbf{x}_1)$ for $h_{\mathbf{A}}$ gives us $\mathbf{A} \cdot (\mathbf{x}_0 - \mathbf{x}_1) = 0$, which implies that we have found a vector $\mathbf{x} = \mathbf{x}_0 - \mathbf{x}_1$ in the kernel of $\mathbf{A}$ that is at most $2t$-sparse. Such a vector can be used to detect bias in LPN samples $(\mathbf{A}, \mathbf{b} = \mathbf{sA} + \mathbf{e})$, since $\mathbf{b} \cdot \mathbf{x} = \mathbf{e} \cdot \mathbf{x}$ is distributed according to the Bernoulli distribution with error probability

$$\epsilon' \leq \frac{1 - (1-\epsilon)^{2t}}{2}, \quad \text{and thus with bias} \quad 1 - 2\epsilon' = (1-\epsilon)^{2t} \geq 2^{-\Omega(\epsilon t)},$$

by the Piling-Up Lemma. For this bias to be noticeable, compared to a uniform random bit $\mathbf{b} \cdot \mathbf{x}$ when $\mathbf{b} \leftarrow \mathbb{F}_2^m$ is sampled randomly, we would need $\epsilon = O(\log n/t) = O(\log^2 n/n)$. With this error probability, LPN is broken in quasi-polynomial time $O(n^{\log n})$. More importantly, we cannot afford a lower error probability so that $\epsilon t = O(1)$, as LPN is fully broken with $\epsilon = O(\log n/n)$.[8]

Achieving Higher Noise Rate with Sparse LPN. Can we hope to achieve collision-resistance with larger error probability? Our key idea is that by changing the distribution of the matrix $\mathbf{A}$, we are able to *reduce* the size of the output space, making compression possible at lower sparsity t (and hence collision-resistance at higher noise rate ϵ). Indeed, if the matrix $\mathbf{A}$ is *sparse* with each column having exactly k ones, where k is a constant or slightly super-constant, then the output space only consists of vectors $\mathbf{y} = \mathbf{Ax} \in \mathbb{F}_2^n$ that are at most kt-sparse. Thus, we achieve compression when

$$\text{number of inputs for } h_{\mathbf{A}} = \binom{m}{t} \quad \gg \quad \sum_{s=0}^{kt} \binom{n}{s} = \text{number of outputs for } h_{\mathbf{A}}.$$

Using binomial approximations and some straightforward calculations, the above is satisfied when

$$\left(\frac{m}{t}\right)^t \gg kt \cdot \left(\frac{en}{kt}\right)^{kt}, \text{ which is implied by } \quad t^{k-1} \gg \Omega\left(\frac{n^k}{m}\right).$$

Thus, as soon as $t^{k-1} \gg \frac{n^k}{m}$, we will have $\mathbf{y}$ lose information on a random t-sparse $\mathbf{x}$. Let us try to understand how large t needs to be. In the most conservative regime when $m = n^{1+\rho}$ for a small constant $\rho > 0$, t^{k-1} must be bigger than $n^{k-1-\rho}$, implying $t > n^{1-\frac{\rho}{k-1}}$ which approaches n as ρ approaches 0. In the most aggressive setting when m is close to $n^{\frac{k}{2}}$, t^{k-1} must be bigger than $n^{\frac{k}{2}}$. This yields $t \approx \sqrt{n}$ if k is a large enough constant.

More generally, if we wish to achieve a compression factor $D > 1$, meaning that the output length is a factor of D smaller than the input length, then we would need

$$\left(\frac{m}{t}\right)^t > \left(kt \cdot \left(\frac{en}{kt}\right)^{kt}\right)^D, \text{ which is implied by } \quad t^{D\cdot k-1} = \Omega\left(\frac{n^{D\cdot k}}{m}\right). \tag{2}$$

We will refer to Equation (2) as the *compression* equation. Similar to the above estimate, we can have $t = n^\delta$ be polynomially smaller than n for a polynomial number of samples $m = n^{1+\rho_{k,D}(\delta)}$, where $\rho_{k,D}(\delta)$ is a constant related to δ. This implies that collision-resistance can be achieved at an inverse polynomial error probability $\epsilon = O(1/t) = O(1/n^\delta)$. We call this the *compression regime* of Sparse LPN.

[8] For $\epsilon = O(\log n/n)$, we can choose n random coordinates of $\mathbf{b} = \mathbf{sA} + \mathbf{e}$, which is error-free with noticeable probability, and then solve for $\mathbf{s}$.

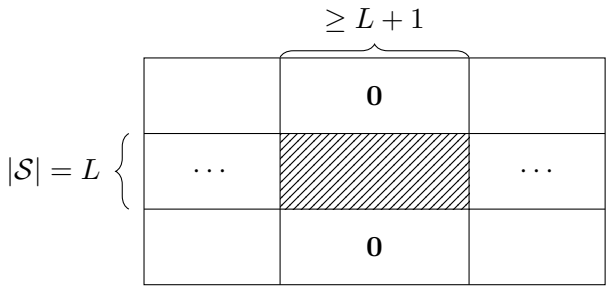

Figure 1: Attack against Sparse LPN with sparsity k, number of samples $m = \Omega(n \cdot (n/t)^{k-1})$, and noise rate $\epsilon = O(\log n/t)$. We focus on a set of rows $\mathcal{S}$ of size $L = O(t)$, and find all columns that is non-zero only within rows contained in T. If we find at least $L+1$ such columns, then we can find a linear dependence between these columns and come up with a $\leq L$-sparse vector $\mathbf{x}$ such that $\mathbf{Ax} = \mathbf{0}$, which can be used to detect noticeable bias in the Sparse LPN samples.

Sparse LPN in its Compression Regime is Broken. Unfortunately, while Sparse LPN in the compression regime would imply collision-resistant hash functions, it is not secure. We give a new but simple attack, that in the compression regime with any factor $D > 1$, one can easily find $O(t)$ sparse vectors $\mathbf{v} \in \mathbb{F}_2^m$ so that $\mathbf{Av} = \mathbf{0}$. Thus, unlike (dense) LPN, Sparse LPN can be broken in polynomial time even at sufficiently small but inverse-polynomial noise probability $\epsilon = O(n^{-\delta})$, and with (related) polynomial number of samples $m = n^{1+\rho_{k,D}(\delta)}$.

The attack can be described as follows. For simplicity, assume that each column of $\mathbf{A}$ is randomly and independently chosen from the set of all k-sparse columns. We want to find a set $\mathcal{S} \subset [n]$ of size L so that there are $L+1$ columns $\{\mathbf{a}_1, \ldots, \mathbf{a}_{L+1}\}$ that are supported entirely in $\mathcal{S}$. Namely, these columns take the value 0 for indices in $[n] \setminus \mathcal{S}$. Once we have found these columns, we can easily find some non-zero combinations combining $\{\mathbf{a}_1, \ldots, \mathbf{a}_{L+1}\}$ that sum to 0 since they must be linearly dependent.

For what L can we expect this to happen? We compute the probability that for an arbitrary set S of size L there exist $L+1$ column vectors supported entirely in S. Since the probability that a single vector is supported in S is $\binom{L}{k}/\binom{n}{k}$, the expected number of such equations becomes roughly $m \cdot \binom{L}{k}/\binom{n}{k}$. Thus, for m samples, we need to set L so that

$$m \cdot \frac{\binom{L}{k}}{\binom{n}{k}} \approx m \cdot \left(\frac{L}{n}\right)^k \gg L, \qquad \text{which is satisfied when } L^{k-1} \approx \frac{n^k}{m}. \tag{3}$$

Therefore, L is up to a constant multiple of t satisfying the compression equation $t^{k-1} = \Omega(\frac{n^k}{m})$.

McEliece-style Wrapper to the Rescue. Note that the above attack crucially relies on being able to identify the support, or locality pattern, of the column vectors. Thus, if we can mask these locality patterns, such as by applying a random linear transformation to the columns, then we can prevent our attack.

This idea of masking a matrix, whose exposure would lead to an efficient attack, goes back to the McEliece cryptosystem [McE78], and is the motivation behind Dense-Sparse LPN. Recall that in McEliece and its variants, the public key consists of a "randomized" code generating matrix $\mathbf{A} = \mathbf{SGP}$, hiding a "nice" representation $\mathbf{G} \in \mathbb{F}_2^{n \times m}$ (such as a binary Goppa code) that enables efficient decoding. Here the masking matrices consist of a random *square* matrix $\mathbf{S} \in \mathbb{F}_2^{n \times n}$ multiplied on the left and a permutation matrix $\mathbf{P} \in \mathbb{F}_2^{m \times m}$ multiplied on the right. The ciphertexts are then LPN samples with respect to this structured matrix.

Our Dense-Sparse LPN assumption makes several modifications to the McEliece template to suit our specific needs. Instead of an algebraically structured matrix $\mathbf{G}$, we work with k-sparse matrices $\mathbf{M}$ which do not have any algebraic structure. We also omit the permutation matrix $\mathbf{P}$ as the sparse matrix distribution can be made invariant under column permutations. Finally, we multiply with a random *compressing* matrix $\mathbf{T} \in \mathbb{F}_2^{\frac{n}{2} \times n}$ to get $\mathbf{A} = \mathbf{TM}$. This change serves two purposes: it is necessary for security, as a square matrix $\mathbf{S} \in \mathbb{F}_2^{n \times n}$ does not adequately mask $\mathbf{M}$ (see Section 7 for the attack), and it aligns with our goal of using $\mathbf{M}$ for its lossy properties rather than for decoding.

We discuss further aspects of Dense-Sparse LPN in Section 2.3; for the moment, we shall go back to constructing collision-resistant hashes from our assumption.

Back to Collision-Resistance. We now re-examine our hash function $h_{\mathbf{A}} = \mathbf{Ax} = \mathbf{TMx}$, which takes t-sparse inputs $\mathbf{x} \in \mathbb{F}_2^m$ and maps them to $\mathbb{F}_2^{\frac{n}{2}}$. When n, m, t is chosen in the compression regime to satisfy Equation (2) for any factor $D > 1$, the mapping $h_{\mathbf{M}}$ sending $\mathbf{x} \mapsto \mathbf{x}' = \mathbf{Mx}$ admits collisions. Since $\mathbf{x}' \in \mathbb{F}_2^n$ is at most kt-sparse, which is lower than the threshold $O(n/\log n)$ for $\mathbf{T}$ to get collisions, it follows that, with overwhelming probability, all collisions of $h_{\mathbf{A}}$ come from collisions of $h_{\mathbf{M}}$.

Unfortunately, $h_{\mathbf{A}}$ is no longer compressing. Since the output of $h_{\mathbf{A}}$ is no longer sparse, we can only bound the output size by its length $\frac{n}{2}$, which is more than the input length of $\log_2 \binom{m}{t} = O(t \log n)$ as t is polynomially smaller than n. To get around this issue, we multiply the function $h_{\mathbf{A}}$ with another compressing matrix $\mathbf{U} \in \mathbb{F}_2^{\ell \times \frac{n}{2}}$ for a suitable ℓ. Equivalently, we now consider the hash family

$$h_{\mathbf{A}'}(\mathbf{x}) = \mathbf{A}' \cdot \mathbf{x} \in \mathbb{F}_2^{\ell}, \qquad \text{where} \quad \mathbf{A}' = \mathbf{U} \cdot \mathbf{A} = (\mathbf{U} \cdot \mathbf{T}) \cdot \mathbf{M} \in \mathbb{F}_2^{\ell \times m}.$$

Note that $\mathbf{V} = \mathbf{U} \cdot \mathbf{T} \in \mathbb{F}_2^{\ell \times n}$ is identically distributed to a random matrix of the same dimensions. We will set ℓ so that two conditions are satisfied:

- To ensure $h_{\mathbf{A}'}$ is compressing, we need $2^{\ell} \ll \binom{m}{t}$.
- To ensure $h_{\mathbf{A}'}$ is collision-resistant, we want the linear transformation $\mathbf{y} \mapsto \mathbf{Vy}$ to map different vectors $\mathbf{y} = \mathbf{Mx}$ that are at most kt-sparse to different elements of $\mathbb{F}_2^{\ell}$. This is satisfied with overwhelming probability if $2^{\ell} \gg \binom{n}{kt}^2 \approx |\mathcal{B}_{\leq}(n, kt)|^2$ by a birthday bound, where $\mathcal{B}_{\leq}(n, kt)$ denotes the Hamming ball of radius kt.

We can meet both conditions by setting the compression factor $D > 2$, so that $\binom{m}{t} \gg \binom{n}{k \cdot t}^2$. We give a formal argument in the full version, showing that collision-resistance can be reduced to Dense-Sparse LPN with an inverse polynomial error probability $\epsilon = O(n^{-\delta})$. Finally, we note that while our later construction of lossy trapdoor functions generically implies collision-resistant hash functions [PW08], the hash construction we just sketched is more direct and conceptually simpler.

2.2 Lossy Trapdoor Functions

We now describe the main ideas behind our lossy trapdoor function construction, which on a high level builds upon the lattice-based template of [PW08]. The core of our contribution lies in identifying that, for the same compression regime of Dense-Sparse LPN as sketched above, we can ensure both lossiness and invertibility depending on the mode being instantiated.

Our function key is of the following form. Given a matrix $\mathbf{A} = \mathbf{TM} \in \mathbb{F}_2^{\frac{n}{2} \times m}$ drawn from the Dense-Sparse LPN matrix distribution, we generate samples $(\mathbf{b}_i = \mathbf{s}_i \mathbf{A} + \mathbf{e}_i)_{i=1}^{\ell}$ for a parameter

$\ell = \ell(n)$ to be chosen later. Equivalently, we may concatenate these samples to get a matrix $\mathbf{B} = \mathbf{SA} + \mathbf{E} \in \mathbb{F}_2^{\ell \times m}$. This matrix $\mathbf{B}$ will be given out as-is for the lossy mode, and for the injective mode, will be used to hide a *special* matrix $\mathbf{C} \in \mathbb{F}_2^{\ell \times m}$ which allows for efficient inversion. Roughly speaking, $\mathbf{C}$ can be seen as a *(robust) compressed sensing* matrix, which can recover a sparse vector $\mathbf{x}$ from a noisy measurement $\mathbf{Cx} + \mathbf{e}$. To summarize, the function key is as follows:

$$\mathsf{fk} = \begin{cases} (\mathbf{A}, \mathbf{B} = \mathbf{SA} + \mathbf{E}) & \text{if } \mathsf{mode} = \mathsf{loss}, \\ (\mathbf{A}, \mathbf{B} = \mathbf{SA} + \mathbf{E} + \mathbf{C}) & \text{if } \mathsf{mode} = \mathsf{inj}, \text{ with } \mathsf{td} = \mathbf{S}. \end{cases}$$

From the Dense-Sparse LPN assumption, it is clear that the two modes are computationally indistinguishable. Now, the input to the function will be t-sparse vectors $\mathbf{x} \in \mathbb{F}_2^m$, and function evaluation returns

$$\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2), \qquad \text{where } \mathbf{y}_1 = \mathbf{Ax} \in \mathbb{F}_2^{\frac{n}{2}}, \text{ and } \mathbf{y}_2 = \mathbf{Bx} \in \mathbb{F}_2^{\ell}.$$

To invert in injective mode (with trapdoor $\mathbf{S}$), we compute $\mathbf{y}' = \mathbf{y}_2 - \mathbf{S}\mathbf{y}_1 = \mathbf{Cx} + \mathbf{Ex}$, then use the decoding guarantee of the compressed sensing matrix $\mathbf{C}$ to recover $\mathbf{x}$ in the presence of noise $\mathbf{Ex}$.

We now analyze the two modes to figure out the parameters that would ensure both lossiness and efficient inversion hold with $1 - \mathsf{negl}(n)$ probability over the choice of the function key.

Lossy Mode. We want to choose parameters so that both $\mathbf{y}_1 = \mathbf{Ax}$ and $\mathbf{y}_2 = (\mathbf{SA} + \mathbf{E})\mathbf{x}$ loses information about $\mathbf{x}$. We will reason separately about these two components as follows:

- Using the decomposition $\mathbf{A} = \mathbf{TM}$ for a random (dense) $\mathbf{T}$ and sparse $\mathbf{M}$, it suffices to have $\mathbf{x}' = \mathbf{Mx}$ lose information about $\mathbf{x}$. This is satisfied in the compression regime for Sparse LPN where Equation (2) requires $t = \Omega\left(n \cdot (n/m)^{\frac{1}{Dk-1}}\right)$ for a compression factor $D > 1$.
- For a fixed value $\mathbf{y}_1$, we can see that $\mathbf{y}_2 = \mathbf{Sy}_1 + \mathbf{Ex}$ lies in a Hamming ball of radius $\|\mathbf{Ex}\|_0$ around $\mathbf{Sy}_1$. If we work with error probability ϵ, then $\mathbf{E} \leftarrow \mathsf{Ber}(\epsilon)^{\ell \times m}$ and $\|\mathbf{x}\|_0 = t$ implies that $\mathbf{Ex} \sim \mathsf{Ber}(\epsilon')^{\ell}$, where $\epsilon' = \frac{1-(1-2\epsilon)^t}{2} \leq \epsilon t$. We want the size of this ball, which is roughly $\binom{\ell}{\epsilon t \ell}$, to be at most a $(D')^{th}$-root of the number of inputs which is $\binom{m}{t}$.

Putting things together, we have that

$$|\{(\mathbf{Ax}, \mathbf{Bx}) \mid \|\mathbf{x}\|_0 = t\}| < \binom{m}{t}^{1/D} \cdot \binom{m}{t}^{1/D'} = \binom{m}{t}^{1/D+1/D'}. \tag{4}$$

Therefore, we can ensure that the output length is an arbitrarily small constant of the input length by setting D, D' to be large enough, with error probability $\epsilon = O(1/t)$ and $\ell = \Theta(\log \binom{m}{t}) = \Theta(t \log n)$.

Injective Mode. Do the parameters required for lossy mode also enable efficient inversion? To answer this, we need to design a matrix $\mathbf{C} \in \mathbb{F}_2^{\ell \times m}$ equipped with an efficient decoding algorithm that can recover a t-sparse vector $\mathbf{x}$ from $\mathbf{y}' = \mathbf{Cx} + \mathbf{e} \in \mathbb{F}_2^{\ell}$, where $\mathbf{e} = \mathbf{Ex}$ is a noise term that is constant-fraction sparse with overwhelming probability.

If our task were to recover a (dense) vector $\mathbf{x}$, then we can simply pick $\mathbf{C}$ to be (the transpose of) an error-correcting code. Then $\mathbf{Cx}$ is (the transpose of) a codeword, which is then perturbed in a constant number of entries to form $\boldsymbol{y}'$. Using an efficient decoding algorithm for $\mathbf{C}$ that can correct a constant fraction of errors, we can recover $\mathbf{x}$ from $\mathbf{y}'$.

However, in our case we have to recover a t-sparse vector $\mathbf{x} \in \mathbb{F}_2^m$. Our idea is to restrict the inversion process to only a *special* subset of such t-sparse vectors, namely the ones that arise as the result of *sparsifying* dense vectors $\mathbf{z} \in \mathbb{F}_2^{t\log(\frac{m}{t})}$. The sparsification process is as follows:

$$\mathbf{z} = [\underbrace{\mathbf{z}_1}_{\log(\frac{m}{t})} \| \dots \| \underbrace{\mathbf{z}_t}_{\log(\frac{m}{t})}] \implies \mathsf{spfy}(\mathbf{z}) = [\underbrace{(0,\dots,1,\dots,0)}_{\mathbf{z}_1\text{-th position}} \| \dots \| \underbrace{(0,\dots,1,\dots,0)}_{\mathbf{z}_t\text{-th position}}] \in \mathbb{F}_2^m,$$

where we interpret $\mathbf{z}_i \in \mathbb{F}_2^{\log(\frac{m}{t})}$ as a number in $\{0,\dots,\frac{m}{t}-1\}$ for all $i \in [t]$. This gives a bijection between binary vectors of length $t\log(\frac{m}{t})$, with *regular* t-sparse vectors of length m. We may also recover $\mathbf{z}$ from $\mathsf{spfy}(\mathbf{z})$ by multiplying with a *gadget matrix* $\mathbf{G} \in \mathbb{F}_2^{t\log(\frac{m}{t})\times m}$ (whose formula can be found in Definition 3.2). In other words, we have $\mathbf{G}\cdot\mathsf{spfy}(\mathbf{z}) = \mathbf{z}$ for all $\mathbf{z} \in \mathbb{F}_2^{t\log(\frac{m}{t})}$. Note that these procedures have appeared in prior works [BLVW19, YZW+19], and can be seen as a code-based analogue of binary decomposition and the gadget matrix in lattice-based cryptography [MP12].

Given these tools, we can instantiate the injective mode as follows. We first slightly change the input space of our function to be regular t-sparse vectors $\mathbf{x} \in \mathbb{F}_2^m$, which are in bijection with $\mathbf{z} = \mathbf{G}\mathbf{x} \in \mathbb{F}_2^{t\log(\frac{m}{t})}$. We now set

$$\mathbf{C} = \mathbf{C}' \cdot \mathbf{G},$$

where $\mathbf{C}' \in \mathbb{F}_2^{\ell\times t\log(\frac{m}{t})}$ is the transpose of an error-correcting code with constant rate and distance [Jus72]. Therefore, using the decoding of $\mathbf{C}'$ we may efficiently recover $\mathbf{z}$, and hence $\mathbf{x} = \mathsf{spfy}(\mathbf{z})$, from $\mathbf{y}' = \mathbf{C}\mathbf{x} + \mathbf{e} = \mathbf{C}'\mathbf{G}\mathbf{x} + \mathbf{e} = \mathbf{C}'\mathbf{z} + \mathbf{e}$.

All-but-one LTDFs. Our LTDF construction also generalize straightforwardly to realizing its *all-but-one (ABO)* variant. In an ABO-LTDF, we have an exponential of branches $\mathcal{B} = \mathbb{F}_2^L$ where one distinguished branch b^* is lossy, and all other branches are injective. Additionally, given the function key fk it is difficult to find out which branch is distinguished. While there is a generic transformation from LTDFs to its all-but-one version [PW08], it presents a tradeoff between the number of branches supported and the degradation of the lossiness parameter.

We avoid this tradeoff by leveraging algebraic properties in our setting. In particular, we achieve ABO-LTDFs with number of branches 2^L, where $L = t\log(\frac{m}{t})$, through a simple twist on our construction above. At a high level, we associate the branches $b \in \mathbb{F}_2^{t\log(\frac{m}{t})}$ with a matrix family $\mathbf{H}_b \in \mathbb{F}_2^{L\times L}$ such that $\mathbf{H}_b - \mathbf{H}_{b'}$ is invertible for all $b \neq b'$. Such a *full-rank difference (FRD)* family of matrices has appeared in prior works [ABB10, KMP14], and we refer to Section 6 for the details of our ABO construction.

Why Low-Noise LPN does not Suffice. As a final remark, let us argue why LPN with noise probability $\epsilon = O(\log^2 n/n)$ does not imply lossy trapdoor functions from our template above. The reason is that since $\epsilon t = O(\log n)$, the vector $\mathbf{E}\mathbf{x}$, with $\mathbf{E} \leftarrow \mathsf{Ber}(\epsilon)^{\ell\times m}$ and $\|\mathbf{x}\|_0 = t$, is Bernoulli distributed with error $\epsilon' = \frac{1}{2} - \frac{1}{\mathsf{poly}(n)}$. This requires setting $\ell = n^{1+\Omega(1)}$ for successful inversion in injective mode.[9] However, such a large ℓ prevents any hope of achieving lossiness, as the Hamming ball around each $\mathbf{y}_1$ is simply too large:

$$\binom{\ell}{\epsilon'\ell} = 2^{n^{1+\Omega(1)}} \gg 2^{O(n)} = \binom{m}{t}. \tag{5}$$

[9] Error-correcting codes of relative distance $\frac{1}{2} - \frac{1}{\mathsf{poly}(n)}$ must have rate at most $1/\mathsf{poly}(n)$, so we have $\ell = n^{1+\Omega(1)}$.

2.3 Discussion on Our Assumption

We now discuss several aspects of our Dense-Sparse LPN assumption. Since our assumption combines aspects of both standard LPN and Sparse LPN, it also inherits some subtle considerations that arise in the context of choosing "good" matrices for Sparse LPN.

Sampling "Good" Matrices for Sparse LPN. In the Sparse LPN assumption, the matrix $\mathbf{M}$ is chosen in $\mathbb{F}_2^{n\times m}$, where $m \ll n^{\frac{k}{2}}$, such that every column is k-sparse where k is some constant. For this parameter setting, there is an inverse polynomial probability of $\mathbf{M}$ having a constant-sparse vector $\mathbf{x}$ in its kernel, so that $\mathbf{Mx} = \mathbf{0}$.[10] When this "bad" event happens, an adversary can find $\mathbf{x}$ in polynomial time and distinguish Sparse LPN samples from random.

However, this "bad" event is only over the choice of matrix $\mathbf{M}$. Outside of this bad event, it is known that if $m = \Omega(n^{1+(\frac{k}{2}-1)(1-\rho)})$ for some $\rho > 0$, then with overwhelming probability, the minimum Hamming weight for a vector $\mathbf{x}$ that satisfies $\mathbf{Mx} = \mathbf{0}$ is at least $\Omega(n^\rho)$ (see Lemma 7.1 for details). Therefore, if we consider Sparse LPN with a large enough noise probability $\epsilon = \Omega(n^{-\rho})$, giving an adversary subexponential time do not seem to increase its advantage beyond the probability of sampling a "bad" matrix.

We note that this issue is present in all prior cryptographic constructions relying on Sparse LPN. Applebaum, Barak and Wigderson [ABW10] resolved this issue by weakening the indistinguishability advantage to be only $o(1)$; this suffices for their application of building public-key encryption, as they could rely on security amplification [HR05]. However, our goal is to build lossy primitives, which do not amplify well [PRS12], so we must rely on a sampling $\mathbf{M}$ from a distribution that avoids sampling a "bad" matrix with overwhelming probability. We will use the following efficiently sampleable distribution from [AK19] which has this property.

Theorem 2.1 (informal). *For every even $k \geq 6$, every $1 < c < k/4$, and every $0 < \gamma < k - 4c$, there exists an efficiently sampleable distribution of k-sparse matrices $\mathbf{M} \in \mathbb{F}_2^{n\times n^c}$ such that with overwhelming probability, every nonzero vector $\mathbf{x}$ in the kernel of $\mathbf{M}$ has Hamming weight at least $O(n^\delta)$, where $\delta = \frac{k-4c-\gamma}{k-\gamma-4}$.*

While this distribution does not sample matrix from all possible parameter regime of Sparse LPN (for instance, we have $m = n^c < n^{k/4}$), it suffices to instantiate the compression regime in Equation (2) with any compression factor $D > 1$, where the noise probability remains inverse polynomial and we have plausible security against subexponential time attacks. Therefore, we can use this distribution to instantiate the sparse matrix $\mathbf{M}$ in our Dense-Sparse LPN assumption.

Almost-Subexponential Regime. Alternatively, we can work with a k that is mildly superconstant, such as $k = \log\log n$. In this setting, $m = n^c$ can be an arbitrary polynomial in n, and the "bad" matrix probability is now negligible, namely $n^{-O(\log\log n)}$. This allows us to sidestep the need for special distributions, and just sample $\mathbf{M}$ uniformly at random from the set of k-sparse matrices. Equation (2) now implies that we can achieve compression factor $D > 1$ when

$$t = O\left(\left(\frac{n^{Dk}}{m}\right)^{\frac{1}{Dk-1}}\right) = n^{1-O\left(\frac{1}{\log\log n}\right)}. \tag{6}$$

Thus, in our construction of LTDFs, we can rely on Dense-Sparse LPN with error probability $\epsilon = O(1/t) = n^{O\left(\frac{1}{\log\log n}\right)-1}$ which is *inverse almost-subexponential*. Note that this noise probability

[10] For instance, $\mathbf{M}$ has two identical columns with probability $O(\frac{m^2}{n^k})$.

is still larger than the $O(\frac{\log^2 n}{n})$ noise rate of (standard) LPN. Therefore, our Dense-Sparse LPN assumption with this error rate is plausibly secure against *almost-subexponential* $2^{n^{O\left(\frac{1}{\log \log n}\right)}}$-time attacks, unlike the quasi-polynomial security of (standard) LPN in its compression regime.

Cryptanalysis. We discuss several attack strategies against Dense-Sparse LPN in Section 7; here we provide a brief summary of their complexity. Let's say we are given a Dense-Sparse LPN instance $(\mathbf{TM}, \mathbf{sTM} + \mathbf{e})$ of dimension n, number of samples $m = n^{1+(k/2-1)(1-\delta)}$, and noise rate ϵ. The first strategy uses generic attacks against LPN (such as information-set decoding [Pra62]), which has time complexity $2^{\widetilde{O}(n \cdot \epsilon)}$. In the second attack, we attempt to find a $\widetilde{O}(n^\delta)$-sparse vector in the kernel of $\mathbf{TM}$, and use it to distinguish Dense-Sparse LPN samples from random; this attack has time complexity $2^{\widetilde{O}(n^\delta)}$ based on generic cycle-finding algorithms. A third line of attack attempts to find $\mathbf{T}$ and $\mathbf{M}$ from their product $\mathbf{TM}$, then use the sparse structure of $\mathbf{M}$ to find short cycles (as in Figure 1). We show that natural attempts to do this actually take *exponential* time, which is less efficient than the prior two approaches. Therefore, we may plausibly conjecture that Dense-Sparse LPN is secure against adversaries running up to time

$$\min\left(2^{\widetilde{O}(n^\delta)}, 2^{\widetilde{O}(n \cdot \epsilon)}\right).$$

A Stronger Variant of Our Assumption. Finally, we can consider making a stronger assumption that together with (standard) LPN implies Dense-Sparse LPN. In this variant, we now assume that the matrix $\mathbf{A} = \mathbf{TM}$ from the Dense-Sparse matrix distribution is actually *pseudorandom*.

Assumption 2.1 (Dense-Sparse Indistinguishability)**.** *For* $\mathbf{T} \leftarrow \mathbb{F}_2^{\frac{n}{2} \times n}$, $\mathbf{M} \leftarrow \mathbb{F}_2^{n \times m}$ *drawn from a good distribution of* k*-sparse matrices, and* $\mathbf{U} \leftarrow \mathbb{F}_2^{\frac{n}{2} \times m}$*, we have* $\mathbf{TM} \approx_c \mathbf{U}$.

By a simple hybrid argument, Assumption 2.1 together with (standard) LPN indeed implies Dense-Sparse LPN with noise rate as small as $O(\frac{\log^2 n}{n})$. We do not see any better attack against this assumption than against Dense-Sparse LPN, and refer to Section 7 for initial cryptanalysis regarding this assumption.

2.4 Open Questions

At a high level, our Dense-Sparse LPN assumption is designed specifically so that we can harness the lossiness property of the k-sparse matrix $\mathbf{M}$, and do so without losing security by applying a random compressing matrix $\mathbf{T}$ before giving out $\mathbf{A} = \mathbf{TM}$. In our construction of collision-resistant hash functions (CRHFs), we even managed to convert this lossiness into actual compression of the output. However, it seems difficult to achieve compression directly without giving up on other properties of the assumption.

Let us explain this difficulty by attempting to construct a (single-server) private information retrieval (PIR) scheme [CGKS95, KO97].[11][12] In PIR, we have a server holding a database $\mathbf{D} \in \mathbb{F}_2^n$, and a client who wishes to learn the i-th entry $\mathbf{D}_i$ of $\mathbf{D}$ without revealing the index i. A client will send a query $\mathbf{q}$ to the server, who then responds with a response $\mathbf{r}$. A non-trivial PIR scheme requires *compactness*, namely that the response length is less than the database length.

[11] It is not known whether lossy trapdoor functions imply PIR, nor is there a black-box separation between two primitives.

[12] Our discussion also presents challenges toward achieving homomorphic primitives from Dense-Sparse LPN. This is because additively homomorphic encryption is known to imply PIR [KO97].

We now consider the following candidate PIR scheme from Dense-Sparse LPN, which borrows from our construction of LTDFs and the template in [KO97]. Assume for simplicity that the database $\mathbf{D} \in \mathbb{F}_2^m$ is t-sparse (the general case can be reduced to this setting by sparsification). To make a query on index $i \in [m]$, the client will send an encryption

$$\mathbf{q} = (\mathbf{A}, \mathbf{b} = \mathbf{s}\mathbf{A} + \mathbf{e} + \mathbf{u}_i), \text{ where } \mathbf{A} = \mathbf{T}\mathbf{M} \in \mathbb{F}_2^{\frac{n}{2} \times m} \text{ and } \mathbf{u}_i = \underbrace{(0, \ldots, 1, \ldots, 0)}_{i\text{-th position}}.$$

The server will respond with the ciphertext applied to $\mathbf{D}$, i.e. with $\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2) = (\mathbf{A}\mathbf{D}, \mathbf{b}\mathbf{D}) \in \mathbb{F}_2^{\frac{n}{2}+1}$. The client then recovers $\mathbf{D}_i = \langle \mathbf{D}, \mathbf{u}_i \rangle$ by computing $\mathbf{r}_2 - \mathbf{s}\mathbf{r}_1 = \mathbf{D}\mathbf{u}_i + \mathbf{D}\mathbf{e}$. By sampling $\mathbf{e}$ from a Bernoulli distribution of probability $\epsilon = O(1/t)$, we can guarantee that the scheme has constant correctness, which can be amplified to negligible by $\omega(\log n)$ repetitions of $\mathbf{b}$. Client's query privacy also follows directly from Dense-Sparse LPN with noise rate ϵ.

The problem with the scheme above is that it is not compact. Indeed, the response is of length $\frac{n}{2} + 1$ which is greater than the database length $\log \binom{m}{t} \approx t \log(m/t)$. Here, we are in a similar situation to our CRHF construction, but we *cannot* use the same trick to achieve compactness here. The reason is that, if we were to multiply $\mathbf{r}_1 = \mathbf{A}\mathbf{D}$ by a further compressing matrix $\mathbf{U} \in \mathbb{F}_2^{\ell \times \frac{n}{2}}$, then the client would receive $\mathbf{U}\mathbf{A}\mathbf{D}$. From there, the client must "decompress" $\mathbf{A}\mathbf{D}$ from $\mathbf{U}$, but this appears computationally infeasible since $\mathbf{U}$ does not come with a trapdoor for efficient inversion!

We note that the same difficulty above also prevents us from using Dense-Sparse LPN to build laconic oblivious transfer and identity-based encryption [BLSV18]. Therefore, we leave as open question the task of overcoming these limitations, either by finding a way to efficiently compress and decompress $\mathbf{A}\mathbf{x}$ for a random t-sparse $\mathbf{x}$, or by proposing a different LPN variant that avoids this issue entirely.

Finally, we also leave finding better attacks on our assumption as an important open question. We also think that it might be plausible to find reductions from well-studied variants of LPN (such as standard LPN or sparse LPN) to our assumption. We leave that as a a great open question.

3 Preliminaries

Notation. Let $\mathbb{N} = \{1, 2, \ldots\}$ be the natural numbers, and define $[a, b] := \{a, a+1, \ldots, b\}$, $[n] := [1, n]$. Our logarithms are in base 2. For a finite set S, we write $x \leftarrow S$ to denote uniformly sampling x from S. We denote the security parameter by λ; our parameters depend on λ, e.g. $n = n(\lambda)$, and we often drop the explicit dependence.

We abbreviate PPT for probabilistic polynomial-time. Our adversaries are non-uniform PPT, or equivalently, polynomial-sized, ensembles $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda \in \mathbb{N}}$. We write $\mathsf{negl}(\lambda)$ to denote a negligible function in λ.

Two ensembles of distributions $\{\mathcal{D}_\lambda\}_{\lambda \in \mathbb{N}}$ and $\{\mathcal{D}'_\lambda\}_{\lambda \in \mathbb{N}}$ are $(T(\lambda), \delta(\lambda))$-*computationally indistinguishable* if any non-uniform PPT adversary $\mathcal{A}$ running in time at most T can distinguish between the two distributions with probability at most δ. Without further specification, we assume the default values of $T(\lambda) = \mathsf{poly}(\lambda)$ and $\delta(\lambda) = \mathsf{negl}(\lambda)$ for indistinguishability. In this default setting, we denote computational indistinguishability by $\{\mathcal{D}_\lambda\}_{\lambda \in \mathbb{N}} \approx_c \{\mathcal{D}'_\lambda\}_{\lambda \in \mathbb{N}}$.

For $q \in \mathbb{N}$ that is a prime power, we write $\mathbb{F}_q$ to denote the finite field with q elements, and $\mathbb{F}_q^\times$ to denote its non-zero elements. We write vectors and matrices in boldcase, e.g. $\mathbf{v} \in \mathbb{F}_q^m$ and $\mathbf{A} \in \mathbb{F}_q^{n \times m}$. Given $\mathbf{v} \in \mathbb{F}_q^m$, we define the Hamming weight $\mathsf{wt}(\mathbf{v})$, also denoted $\|\mathbf{v}\|_0$, to be the number of non-zero entries of $\mathbf{v}$.

Bernoulli Distribution. We denote the Bernoulli distribution over a finite field $\mathbb{F}_q$ with noise rate $\epsilon \in (0,1)$ by $\mathsf{Ber}(\mathbb{F}_q, \epsilon)$; this distribution gives 0 with probability $1-\epsilon$, and a random non-zero element of $\mathbb{F}_q$ with probability ϵ. We write $e \sim \mathsf{Ber}(\mathbb{F}_q, \epsilon)$ to denote that e comes from the corresponding Bernoulli distribution. When $q = 2$, we omit $\mathbb{F}_q$ and simply write $\mathsf{Ber}(\epsilon)$.

Definition 3.1 (Bias)**.** *Let $\mathbb{F}$ be a finite field. Given a distribution $\mathcal{D}$ over $\mathbb{F}^m$ and a vector $\mathbf{u} \in \mathbb{F}^m$, we define the* bias *of $\mathcal{D}$ with respect to $\mathbf{u}$ to be*

$$\mathsf{bias}_{\mathbf{u}}(\mathcal{D}) := \left| \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}} [\langle \mathbf{u}, \mathbf{x} \rangle] - \frac{1}{|\mathbb{F}|} \right|.$$

The bias *of $\mathcal{D}$ is defined as $\mathsf{bias}(\mathcal{D}) = \max_{\mathbf{u}\neq\mathbf{0}} \mathsf{bias}_{\mathbf{u}}(\mathcal{D})$.*

Lemma 3.1 (Bias of the Bernoulli distribution)**.** *For any finite field $\mathbb{F}_q$, noise rate $\epsilon \in (0,1)$, and $d \in \mathbb{N}$, consider the noise distribution $\mathcal{D}_{m,n} = (\mathsf{Ber}(\mathbb{F}_q, \epsilon))^m$. Then $\epsilon_d = \left(1 - \frac{q}{q-1}\epsilon\right)^d$.*

As a special case when $q = 2$, we have the piling-up lemma.

Lemma 3.2 (Piling-Up Lemma)**.** *For any $\epsilon \in (0,1)$, we have that*

$$\Pr\left[\sum_{i=1}^{\ell} e_i = 1 \;\middle|\; e_1, \ldots, e_\ell \leftarrow \mathsf{Ber}(\epsilon)\right] = \frac{1-(1-2\epsilon)^\ell}{2} < \min\left(\epsilon\ell, \frac{1}{2} - 2^{-4\epsilon\ell - 1}\right).$$

Tail Bounds. We also state some standard tail bounds for binary variables.

Lemma 3.3 (Chernoff/Hoeffding bound)**.** *Let $X_1, \ldots, X_n \in \{0,1\}$ be i.i.d random variables with mean at most ϵ. Then for every $\kappa > 1$,*

$$\Pr[X_1 + \cdots + X_n > (1+\kappa)\epsilon n] \leq e^{-2\kappa^2 \epsilon n}.$$

Binomial Approximation. We will use the following basic approximations of the binomial coefficient, which can be found in e.g. [CLRS22]. Namely, for any $1 \leq k \leq n/2$, we have

$$\left(\frac{n}{k}\right)^k \leq \binom{n}{k} \leq \left(\frac{en}{k}\right)^k. \tag{7}$$

Hamming Balls. Given $n, w \in \mathbb{N}$ with $w \leq n$, we define the following sets:

- $\mathcal{B}_{\leq}(n,w) = \{\mathbf{x} \in \{0,1\}^n \mid \mathsf{wt}(\mathbf{x}) \leq w\}$,
- $\mathcal{B}(n,w) = \{\mathbf{x} \in \{0,1\}^n \mid \mathsf{wt}(\mathbf{x}) = w\}$,
- $\mathcal{B}_{\mathsf{reg}}(n,w) = \{\mathbf{x} = \mathbf{x}_1 \| \ldots \| \mathbf{x}_w \in \{0,1\}^n \mid \mathbf{x}_i \in \{0,1\}^{n/w} \wedge \mathsf{wt}(\mathbf{x}_i) = 1 \; \forall i \in [w]\}$, for any w dividing n.

Their sizes are as follows.

Lemma 3.4. *For any $w < n/3$, we have the following:*

- $|\mathcal{B}_{\leq}(n,w)| = \sum_{t=0}^{w} \binom{n}{t} \in \left(\binom{n}{w}, 2\binom{n}{w}\right)$.
- $|\mathcal{B}(n,w)| = \binom{n}{w} \in \left(2^{w\log(n/w)}, 2^{w\log(en/w)}\right)$.

- $|\mathcal{B}_{\mathsf{reg}}(n,w)| = \binom{n}{w}^w = 2^{w\log(n/w)}$.

Thus, we have that $|\mathcal{B}_{\mathsf{reg}}(n,w)| < |\mathcal{B}(n,w)| < |\mathcal{B}_{\leq}(n,w)| < 2^{w\log e+1}\cdot|\mathcal{B}_{\mathsf{reg}}(n,w)|$.

Proof. The only non-trivial claim is that $\sum_{t=0}^{w}\binom{n}{t} < 2\binom{n}{w}$. This follows from the fact that $\binom{n}{t-1} = \frac{t}{n-t+1}\binom{n}{t} < \frac{1}{2}\binom{n}{t}$ for all $t < n/3$, and hence $\sum_{t=0}^{w}\binom{n}{t} < \left(1+\frac{1}{2}+\left(\frac{1}{2}\right)^2+\ldots\right)\binom{n}{w} = 2\binom{n}{w}$. □

Sparsification and the Gadget Matrix. We describe an LPN analogue to the LWE gadget matrix [MP12], based on the idea of sparsification.

Given any $s\in\mathbb{N}$, we denote by $\mathsf{bin}_s : [0,2^s-1]\to\{0,1\}^s$ the binary decomposition function, and $\mathsf{bin}_s^{-1} : \{0,1\}^s\to[0,2^s-1]$ its inverse. Given any $\mathbf{v}\in\{0,1\}^s$, we define $\mathsf{ind}(\mathbf{v})$ to be the vector of length 2^s (indexed from 0) consisting of all zeros, except for the $\mathsf{bin}_s^{-1}(\mathbf{v})$-th entry being 1.

Definition 3.2. *Let $n,w\in\mathbb{N}$ with $w\leq n$, and define $\widetilde{w} = w\log(n/w)$. Given $\mathbf{x}\in\{0,1\}^{\widetilde{w}}$ divided into w blocks of size $s=\log(n/w)$, so that $\mathbf{x}=\mathbf{x}_1\|\ldots\|\mathbf{x}_w$, we define its (n,w)-*sparsification $\mathsf{spfy}_{n,w}(\mathbf{x})$ *to be*

$$\mathsf{spfy}_{n,w}(\mathbf{x}) := \mathsf{ind}(\mathbf{x}_1)\|\ldots\|\mathsf{ind}(\mathbf{x}_w)\in\{0,1\}^n.$$

*We also define the (n,w)-*gadget matrix $\mathbf{G}_{n,w}$ *to be*

$$\mathbf{G}_{n,w} := \mathbf{g}_s\otimes\mathbf{I}_w\in\{0,1\}^{w\times n},\quad \textit{where } \mathbf{g}_s := [\mathsf{bin}_s(0)\|\mathsf{bin}_s(1)\|\ldots\|\mathsf{bin}_s(2^s-1)]\,,$$

and $\mathbf{I}_w$ is the identity matrix of dimension w. For convenience, we also refer to $\mathsf{spfy}_{n,w}$ as $\mathbf{G}_{n,w}^{-1}$, and omit the subscripts n,w when they are clear from context.

We can easily check that

$$\mathbf{G}_{n,w}\cdot\mathbf{G}_{n,w}^{-1}(\mathbf{x}) = \mathbf{G}_{n,w}\cdot\mathsf{spfy}_{n,w}(\mathbf{x}) = \mathbf{x}. \tag{8}$$

Note that for the dimensions of $\mathbf{G}_{n,w}$ to satisfy $n=\mathsf{poly}(w)$, we need $s\leq c\log w$ for some constant c. We also get the following identity for any $\mathbf{x},\mathbf{y}\in\{0,1\}^{\widetilde{w}}$:

$$\langle\mathbf{x},\mathbf{y}\rangle = \left\langle\mathbf{x},\mathbf{G}_{n,w}\cdot\mathsf{spfy}_{n,w}(\mathbf{y})\right\rangle = \left\langle\mathbf{G}_{n,w}^{\mathsf{T}}\cdot\mathbf{x},\mathsf{spfy}_{n,w}(\mathbf{y})\right\rangle. \tag{9}$$

3.1 Coding Theory

Definition 3.3 (Dual Distance)**.** *Let $\mathbb{F}_q$ be a finite field and $n<m\in\mathbb{N}$. The* dual distance *of a matrix $\mathbf{A}\in\mathbb{F}_q^{n\times m}$, denoted $\mathsf{dd}(\mathbf{A})$, is defined to be minimum sparsity of a vector $\mathbf{x}\in\mathbb{F}_q^m$ in the kernel of $\mathbf{A}$. In other words, we define $\mathsf{dd}(\mathbf{A}) = \min\{\mathsf{wt}(\mathbf{x})\mid\mathbf{A}\mathbf{x}=\mathbf{0}\}$.*

Definition 3.4 (q-ary Entropy)**.** *For any $x\in(0,1)$, we define the q-ary entropy function to be $H_q(x) = x\log_q(q-1) - x\log_q x - (1-x)\log_q(1-x)$. We denote binary entropy by $H(x)$.*

The following standard results can be found in coding theory textbooks, e.g. [GRS12].

Lemma 3.5 (Entropy Approximation)**.** *For any $\delta\in(0,1)$, we have that $\delta\log(1/\delta) < H(\delta) < \delta\log(4/\delta)$.*

Lemma 3.6 (Gilbert-Varshamov Bound)**.** *Let $\mathbb{F}_q$ be a finite field. For every $\delta\in(0,1-1/q)$ and every $\epsilon\in(0,1-H_q(\delta))$, letting $k=\lfloor(1-H_q(\delta)-\epsilon)\cdot n\rfloor$, a random matrix $\mathbf{G}\leftarrow\mathbb{F}_q^{k\times n}$ generates a q-ary linear code of distance at least δn with probability at least $1-q^{-\epsilon n}$.*

Equivalently, for $\ell=\lceil(H_q(\delta)+\epsilon)\cdot n\rceil$, a random matrix $\mathbf{H}\leftarrow\mathbb{F}_q^{\ell\times n}$ is a parity-check matrix for a q-ary linear code of distance at least δn with probability at least $1-q^{-\epsilon n}$.

Lemma 3.7 (Asymptotically Good Codes)**.** *For some constants $\delta,\rho>0$, there exists an explicit construction of a family of binary codes $\{\mathbf{C}_n\}_{n\in\mathbb{N}}$ with block length $N(n)$ (bounded by some fixed polynomial in n), rate ρ, and supporting efficient error correction for up to δn errors.*

4 Our Code-Based Assumption: Dense-Sparse LPN

In this section, we define our main assumption, Dense-Sparse LPN. To do so, we will first define a general LPN assumption with an arbitrary distribution of coefficient matrix.

Definition 4.1 (Decisional $(\mathcal{M}, \epsilon)$-LPN)**.** *Let $n \in \mathbb{N}$ be the dimension, $m = m(n)$ be the number of samples, $\epsilon = \epsilon(n) \in (0,1)$ be the noise rate, and $q = q(n)$ be a prime power. Given an efficiently sampleable distribution $\mathcal{M} = \mathcal{M}(n, m, \mathbb{F}_q)$ over matrices in $\mathbb{F}_q^{n \times m}$, we say that the* $(\mathcal{M}, \epsilon)$-LPN assumption *is $(T(n), \delta(n))$-hard if for all adversary $\mathcal{A}$ running in time at most T, the following holds:*

$$\mathbf{Adv}^{\mathsf{LPN}}_{n,m,q,\mathcal{M},\epsilon}(\mathcal{A}) := \left| \begin{array}{l} \Pr\left[\mathcal{A}(\mathbf{A}, \mathbf{sA} + \mathbf{e}) = 1 \;\middle|\; \mathbf{A} \leftarrow \mathcal{M}, \mathbf{s} \leftarrow \mathbb{F}_q^{1\times n}, \mathbf{e} \leftarrow \mathsf{Ber}(\mathbb{F}_q, \epsilon)^{1\times m}\right] \\ - \Pr\left[\mathcal{A}(\mathbf{A}, \mathbf{u}) = 1 \;\middle|\; \mathbf{A} \leftarrow \mathcal{M}, \mathbf{u} \leftarrow \mathbb{F}_q^{1\times m}\right] \end{array} \right| \leq \delta.$$

*We say that $(\mathcal{M}, \epsilon)$-*LPN *is* polynomially hard *if for every polynomial $p(n)$, there exists a negligible function $\mathsf{negl}(n)$ such that it is $(p(n), \mathsf{negl}(n))$-hard. Similarly, $(\mathcal{M}, \epsilon)$-*LPN *is* subexponentially hard *if there exists a constant $0 < c < 1$ and a negligible function $\mathsf{negl}(n)$ such that it is $(2^{n^c}, \mathsf{negl}(n))$-hard.*

Definition 4.2 (Decisional $(\mathcal{M}, \epsilon)$-Dual LPN)**.** *Consider $n, m, q, \mathcal{M}, \epsilon$ as in Definition 4.1. We say that the* $(\mathcal{M}, \epsilon)$-dual LPN assumption *is $(T(n), \delta(n))$-hard if for all adversary $\mathcal{A}$ running in time at most T, the following holds:*

$$\mathbf{Adv}^{\mathsf{dual\text{-}LPN}}_{n,m,q,\mathcal{M},\epsilon}(\mathcal{A}) := \left| \begin{array}{l} \Pr\left[\mathcal{A}(\mathbf{H}, \mathbf{He}) = 1 \;\middle|\; \mathbf{H} \leftarrow \mathcal{M}, \mathbf{e} \leftarrow \mathsf{Ber}(\mathbb{F}_q, \epsilon)^{1\times m}\right] \\ - \Pr\left[\mathcal{A}(\mathbf{H}, \mathbf{u}) = 1 \;\middle|\; \mathbf{H} \leftarrow \mathcal{M}, \mathbf{u} \leftarrow \mathbb{F}_q^{1\times m}\right] \end{array} \right| \leq \delta.$$

*We may define polynomial or subexponential hardness of $(\mathcal{M}, \epsilon)$-*dual-LPN *similar to Definition 4.1.*

Remark 4.1. For the rest of the paper, we will work over the binary field (so that $q = 2$). We note that our assumptions and constructions can be straightforwardly generalized to work with any constant $q = O(1)$.

When $\mathcal{M}$ is the uniform distribution over $\mathbb{F}_2^{n\times m}$, we recover the (standard) LPN assumption [BFKL94] (with Dual-LPN equivalent to LPN). We now define variants of LPN with different matrix distributions $\mathcal{M}$.

Definition 4.3 (Sparse LPN)**.** *Let $k \in \mathbb{N}$ be a constant, and consider parameters $n \in \mathbb{N}$, $m = m(n) < n^{k/2}$, and $\epsilon = \epsilon(n) < 1$. Denote by $\mathsf{SpMat}(n, m, k)$ the set of matrices $\mathbf{A} \in \mathbb{F}_2^{n\times m}$ such that each column of $\mathbf{A}$ has exactly k non-zero entries.*

*We define the (n, m, k, ϵ)-*sparse LPN (sLPN) assumption *to be the following: there exists an efficiently sampleable distribution $\mathcal{M}_{\mathsf{sp}}$ over $\mathsf{SpMat}(n, m, k)$ such that the $(\mathcal{M}_{\mathsf{sp}}, \epsilon)$-*LPN *assumption holds.*

Note that the above assumption does not specify the exact distribution $\mathcal{M}_{\mathsf{sp}}$ of sparse matrices. This is because the "canonical" distribution $\mathcal{M}_{\mathsf{unif}}$ of sampling k-sparse columns independently and uniformly at random does *not* suffice for the above assumption. The reason is that there is a noticeable probability, of roughly $O(m^2/n^k)$, for sampling a "bad" matrix $\mathbf{A} \leftarrow \mathcal{M}_{\mathsf{unif}}$ with small dual distance, which would break the assumption by giving the adversary a noticeable distinguishing advantage.

Motivated by this discussion, we lay out a necessary criteria for the sparse matrix distribution $\mathcal{M}_{\mathsf{sp}}$ to satisfy the Sparse LPN assumption.

Definition 4.4 (Sparse matrices with $\omega(1)$-dual distance)**.** *For every $n \in \mathbb{N}$, $k = k(n)$, $m = m(n) < n^{k/2}$ and $d = \omega(1)$, define $\mathsf{SpMat}(n, m, k, d) = \{\mathbf{A} \in \mathsf{SpMat}(n, m, k) \mid \mathsf{dd}(\mathbf{A}) \geq d\}$ to be the subset of $\mathsf{SpMat}(n, m, k)$ consisting of matrices with super-constant dual distance of at least d.*

*We say that an efficiently sampleable distribution $\mathcal{M}_{\mathsf{sp}}$ over $\mathsf{SpMat}(n, m, k)$ is (d, δ)-*good *if $\mathcal{M}_{\mathsf{sp}}$ has min-entropy at least n^c for some constant $c > 0$, and furthermore,*

$$\Pr[\mathbf{A} \notin \mathsf{SpMat}(n, m, k, d) \mid \mathbf{A} \leftarrow \mathcal{M}_{\mathsf{sp}}] \leq \delta.$$

We say that $\mathcal{M}_{\mathsf{sp}}$ is good *if it is (d, δ)-good for some $d = \omega(1)$ and $\delta = \mathsf{negl}(n)$.*

In the definition above, we require the distribution $\mathcal{M}_{\mathsf{sp}}$ to have sufficient min-entropy; this is to prevent a guessing attack from a non-uniform adversary. We may make the *conjecture* that any such good distribution $\mathcal{M}_{\mathsf{sp}}$ would give rise to a secure sLPN assumption. A more conservative assumption would be to assume sLPN assumption holds for *specific* good distributions that have been constructed in the literature. In our work, we will use the following distribution by Applebaum and Kachlon [AK19].

Theorem 4.1 (Theorem 7.18 [AK19], adapted)**.** *For every even $k \geq 6$, every $1 < c < k/4$ with $\gamma = k - 4c$, there exists an efficiently computable, $\left(O(n^{\delta}), n^{-O\left(\frac{\log\log\log n}{\log\log\log\log n}\right)}\right)$-good distribution over $\mathsf{SpMat}(n, m, k)$, where $m = n^c$ and $\delta = \frac{k-4c-\gamma}{k-\gamma-4}$. We call this the* AK19 distribution*.*

Note that the AK19 distribution does not give us all possible parameter range for k-sparse matrices. In particular, the number of samples m is limited to be at most $n^{k/4}$. However, the parameter regime that the AK19 distribution supports overlap with the compression regime of Lemma 4.1 for any constant compression factor $D > 1$. Therefore, we can indeed use this distribution to instantiate our schemes and assumptions.

Definition 4.5 (Dense-Sparse LPN)**.** *Let $k \in \mathbb{N}$, $\alpha \in (0, 1)$ be constants, and consider parameters $n \in \mathbb{N}$, $m = m(n) < n^{k/2}$, and $\epsilon = \epsilon(n) < 1$. Let $\mathcal{M}_{\mathsf{sp}}$ be a good distribution over $\mathsf{SpMat}(n, m, k)$. We define the $(n, m, k, \mathcal{M}_{\mathsf{sp}}, \epsilon)$-*Dense-Sparse LPN (DS-LPN) *assumption to be the $(\mathcal{M}_{\mathsf{DS}}, \epsilon)$-LPN assumption, where $\mathcal{M}_{\mathsf{DS}}$ is the following distribution:*

$$\mathcal{M}_{\mathsf{DS}} = \{\mathbf{T} \cdot \mathbf{M} \mid \mathbf{T} \leftarrow \mathbb{F}_2^{\alpha n \times n}, \mathbf{M} \leftarrow \mathcal{M}_{\mathsf{sp}}\}.$$

In other words, we say that Dense-Sparse LPN is $(T(n), \delta(n))$-hard if the following holds for every adversary $\mathcal{A}$ running in time at most T:

$$\mathbf{Adv}^{\mathsf{DS\text{-}LPN}}_{n,m,k,\mathcal{M}_{\mathsf{sp}},\epsilon}(\mathcal{A}) := |\Pr[\mathcal{A}(\mathbf{A}, \mathbf{sA} + \mathbf{e}) = 1] - \Pr[\mathcal{A}(\mathbf{A}, \mathbf{u}) = 1]| \leq \delta,$$

where $\mathbf{T} \leftarrow \mathbb{F}_2^{\alpha n \times n}$, $\mathbf{M} \leftarrow \mathcal{M}_{\mathsf{sp}}$, $\mathbf{A} = \mathbf{T} \cdot \mathbf{M}$, $\mathbf{s} \leftarrow \mathbb{F}_2^{1 \times \alpha n}$, $\mathbf{e} \leftarrow \mathsf{Ber}(\epsilon)^{1 \times m}$, and $\mathbf{u} \leftarrow \mathbb{F}_2^m$.

To simplify notation, we will take $\alpha = 1/2$ for all of our constructions, though the assumption plausibly holds for any constant $\alpha \in (0, 1)$. We provide detailed cryptanalysis of Dense-Sparse LPN in Section 7.

Remark 4.2 (Dense-Sparse Dual-LPN)**.** It is plausible that the dual version of Dense-Sparse LPN also holds, namely that

$$\{(\mathbf{TM}, \mathbf{TMe})\} \approx_c \{(\mathbf{TM}, \mathbf{u})\},$$

for $\mathbf{T} \leftarrow \mathbb{F}_2^{\alpha n \times n}$, $\mathbf{M} \leftarrow \mathcal{M}_{\mathsf{sp}}$, $\mathbf{e} \leftarrow \mathsf{Ber}(\epsilon)^{m \times 1}$, and $\mathbf{u} \leftarrow \mathbb{F}_2^{\alpha n \times 1}$. This is because $\mathbf{Me}$ is roughly $O(kt)$-sparse, so that plausibly $(\mathbf{T}, \mathbf{TMe}) \approx_c (\mathbf{T}, \mathbf{u})$ by dual-LPN for the random matrix $\mathbf{T} \in \mathbb{F}_2^{\alpha n \times n}$.

Note that this is not yet a rigorous argument since the sparse vector $\mathbf{Me}$ is very far from being Bernoulli distributed. Since we do not use Dense-Sparse dual-LPN in our constructions, we leave a more rigorous cryptanalysis and further applications of this assumption to future work.

Compression Regime. We give here the parameter regime that allows for the function

$$f_{\mathbf{M}} : \mathcal{B}_{\mathsf{reg}}(m,t) \to \mathcal{B}_{\leq}(n,kt) \quad \text{defined by} \quad f_{\mathbf{M}}(\mathbf{x}) = \mathbf{M}\cdot\mathbf{x},$$

for any $\mathbf{M} \in \mathsf{SpMat}(n,m,k)$, to be compressing (by some constant factor D in the exponent).

Lemma 4.1. *Given constant $k \in \mathbb{N}, k \geq 3$ and compression factor $D > 1$, define $\delta_{(4.1)}(k,D) := 1 - \frac{k/2-1}{Dk-1}$. For any $\delta \in (\delta_{(4.1)}(k,D), 1)$, any $n \in \mathbb{N}$ large enough, and any $m < n^{k/2}$, letting $t = n^{\delta}$, we have the following:*

$$|\mathcal{B}_{\mathsf{reg}}(m,t)| = 2^{t\log(m/t)} > \left(2^{kt\log(en/kt)+1}\right)^{D} > |\mathcal{B}_{\leq}(n,kt)|^{D} \tag{10}$$

whenever $m \geq m_{(4.1)}(n,k,D,\delta) := n^{1+(Dk-1)(1-\delta)}$.

Proof. Note that the last inequality in Equation (10) follows from Lemma 3.4. Thus it suffices to show the central inequality. Taking the logarithms of both sides, we need to show that

$$t(\log m - \log t) > D\left(kt\log(en/kt) + 1\right).$$

Dividing by t and isolating out $\log(m)$, we get

$$\log m > D/t + \log(t) + Dk\log(en/kt),$$

which is equivalent to

$$m > 2^{D/t}\cdot t\cdot\left(\frac{en}{kt}\right)^{Dk} = 2^{D/t}\cdot\left(\frac{e}{k}\right)^{Dk}\cdot\frac{n^{Dk}}{t^{Dk-1}}.$$

Plugging in $t = n^{\delta}$ gives us

$$m > 2^{D/n^{\delta}}\cdot\left(\frac{e}{k}\right)^{Dk}\cdot n^{1+(Dk-1)(1-\delta)}.$$

Since $\lim_{n\to\infty} 2^{D/n^{\delta}} = 1$ and $e/k \leq e/3 < 1$, for n large enough we have

$$2^{D/n^{\delta}}\cdot\left(\frac{e}{k}\right)^{Dk} < \left(\frac{e}{k}\right)^{-Dk}\cdot\left(\frac{e}{k}\right)^{Dk} = 1.$$

Therefore, Equation (10) is satisfied when $m \geq n^{1+(Dk-1)(1-\delta)}$. Finally, since we need to impose the condition that $m < n^{k/2}$, it follows that δ is at most $1 - \frac{k/2-1}{Dk-1} = \delta_{(4.1)}(k,D)$. □

5 Collision-Resistant Hash Functions

In this section, we give a simple construction of collision-resistant hash functions (CRHFs) from our Dense-Sparse LPN assumption. While lossy trapdoor functions (LTDFs), which we construct in Section 6, are known to imply CRHFs [PW08], the resulting construction is more complex than the one presented here (and require a slightly smaller noise rate $\epsilon = O(1/n^{\delta})$ instead of $\epsilon = O(\log n/n^{\delta})$).

Compared to prior LPN-based hashes [BLVW19, YZW+19], which rely on either inverse quasi-polynomial noise rate $\epsilon = O(\log^2 n/n)$ or sub-exponential hardness, our construction only requires polynomial hardness and inverse-polynomial noise rate $\epsilon = \widetilde{O}(n^{-\delta})$ for DS-LPN, where δ is a constant that can be arbitrarily close to $3/4$ (as $k \to \infty$).

Collision-Resistant Hash Function from Dense-Sparse LPN

Parameters.

- Pick any constant $k \in \mathbb{N}, k \geq 3$, any constant $D > 2$, and any $\delta \in (\delta_{(4.1)}(k, D), 1)$. Let $\rho = 1/2 - 1/D > 0$.
- Let $m > m_{(4.1)}(n, k, D, \delta)$, $t = n^\delta$, and $\epsilon = O(\log n/t)$. Denote $\widetilde{t} = t \log(m/t)$. Choose $n = (2\lambda/\rho)^{1/\delta}$ so that $\rho\widetilde{t} > \rho n^\delta > 2\lambda$. Let $\mathcal{M}_{\mathsf{sp}}$ be a good distribution over $\mathsf{SpMat}(n, m, k)$.

Construction.

- $\mathsf{Gen}(1^\lambda) \to \mathsf{k}$. Sample $\mathbf{H} \leftarrow \mathbb{F}_2^{(1-\rho)\widetilde{t} \times n}$, and $\mathbf{M} \leftarrow \mathcal{M}_{\mathsf{sp}}$. Return $\mathsf{k} = \mathbf{A}' = \mathbf{H} \cdot \mathbf{M}$.
- $\mathsf{Hash}(\mathsf{k}, \mathbf{x}) \to \mathsf{h}$. On input $\mathbf{x} \in \mathbb{F}_2^{\widetilde{t}}$, return $\mathsf{h} = \mathbf{A}' \cdot \mathsf{spfy}_{m,t}(\mathbf{x}) \in \mathbb{F}_2^{(1-\rho)\widetilde{t}}$.

Figure 2: CRHF construction

Definition 5.1. *A* collision-resistant hash function (CRHF) *family, with input length $m(\lambda)$ and output length $n(\lambda)$, is a tuple of PPT algorithms $\mathcal{H} = (\mathsf{Gen}, \mathsf{Hash})$ with the following properties:*

- ***Syntax:***
  - $\mathsf{Gen}(1^\lambda) \to \mathsf{k}$. *On input the security parameter 1^λ, output a hash key* k.
  - $\mathsf{Hash}(\mathsf{k}, \mathbf{x})$. *On input the hash key* k *and an input $\mathbf{x} \in \{0,1\}^{m(\lambda)}$, deterministically output $\mathsf{h} \in \{0,1\}^{n(\lambda)}$.*
- ***Compression:*** *We have $m(\lambda) > n(\lambda) + 2\lambda$ for all $\lambda \in \mathbb{N}$.*
- ***Collision-Resistance:*** *For all polynomial-size adversary $\mathcal{A}$, the following probability is negligible:*

$$\mathbf{Adv}^{\mathsf{CRHF}}(\mathcal{A}) := \Pr\left[\begin{array}{c|l} \mathbf{x}_1 \neq \mathbf{x}_2 \quad \wedge & \mathsf{k} \leftarrow \mathsf{Gen}(1^\lambda) \\ \mathsf{Hash}(\mathsf{k}, \mathbf{x}_1) = \mathsf{Hash}(\mathsf{k}, \mathbf{x}_2) & (\mathbf{x}_1, \mathbf{x}_2) \leftarrow \mathcal{A}(\mathsf{k}) \end{array}\right] \leq \mathsf{negl}(\lambda).$$

Theorem 5.1. *Assuming the $(n, m, k, \mathcal{M}_{\mathsf{sp}}, \epsilon)$-DS-LPN assumption holds, where $n, m, k, \mathcal{M}_{\mathsf{sp}}, \epsilon$ are defined as in Figure 2. Then Figure 2 gives a construction of a CRHF family.*

Proof. It is clear that we have compression, as the gap between input and output size is $\widetilde{t} - (1-\rho)\widetilde{t} = \rho\widetilde{t} > 2\lambda$ by our parameter choice. It remains to show collision-resistance, which we show by going through the following hybrids.

- Hyb_0: this is the CHRF game, where an adversary $\mathcal{A}$ receives $\mathbf{A}' = \mathbf{H} \cdot \mathbf{M}$ and outputs $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{F}_2^{t \log(m/t)}$. $\mathcal{A}$ wins, or equivalently Hyb_0 returns 1, if $\mathbf{x}_1 \neq \mathbf{x}_2$ and $\mathbf{A}' \cdot \mathsf{spfy}_{m,t}(\mathbf{x}_1) = \mathbf{A}' \cdot \mathsf{spfy}_{m,t}(\mathbf{x}_2)$; equivalently, when $\mathbf{A}' \cdot \mathbf{x}' = \mathbf{0}$ where $\mathbf{x}' = \mathsf{spfy}_{m,t}(\mathbf{x}_1) - \mathsf{spfy}_{m,t}(\mathbf{x}_2)$.
- Hyb_1: this is identical to Hyb_0, except the adversary $\mathcal{A}$ only wins if $\mathbf{M} \cdot \mathbf{x}' = \mathbf{0}$ (and also $\mathbf{x}_1 \neq \mathbf{x}_2$). Thus, Hyb_2 differs from Hyb_1 only when $\mathbf{M} \cdot \mathbf{x}' \neq \mathbf{0}$ but $\mathbf{H} \cdot (\mathbf{M} \cdot \mathbf{x}') = \mathbf{0}$. Since $\mathbf{y} = \mathbf{M} \cdot \mathbf{x}' \in \mathbb{F}_2^n$ is at most $2kt$-sparse, this amounts to bounding the probability that a random parity-check matrix $\mathbf{H} \in \mathbb{F}_2^{(1-\rho)\widetilde{t} \times n}$ has distance less than $2kt$.

We will show that this probability is negligible by examining our parameter choices. By Lemma 3.5, we have that

$$H\left(\frac{2kt}{n}\right) \cdot n < 2kt \log\left(\frac{2n}{kt}\right).$$

Since we choose $t = n^\delta$ and $m > m_{(4.1)}(n, k, D, \delta)$, we know from Lemma 4.1 that the following holds:

$$\begin{aligned} & \widetilde{t} = t \log\left(\frac{m}{t}\right) > D\left(kt \log\left(\frac{en}{kt}\right) + 1\right) > Dkt \log\left(\frac{2n}{kt}\right) \\ \implies & \frac{2}{D}\widetilde{t} > 2kt \log\left(\frac{2n}{kt}\right) > H\left(\frac{2kt}{n}\right) n \\ \implies & (1-\rho)\widetilde{t} = \left(\frac{2}{D} + \rho\right)\widetilde{t} > H\left(\frac{2kt}{n}\right) n + 2\lambda. \end{aligned}$$

Therefore, by Gilbert-Varshamov (Lemma 3.6), $\mathbf{H} \in \mathbb{F}_2^{(1-\rho)\widetilde{t} \times n}$ has distance less than $2kt$ with probability at most $2^{-2\lambda} = \mathsf{negl}(\lambda)$.

- Hyb_2: this is identical to Hyb_1, but instead of sampling $\mathbf{H} \leftarrow \mathbb{F}_2^{2t\log(m/t) \times n}$, we sample $\mathbf{H} = \mathbf{H}' \cdot \mathbf{T}$ where $\mathbf{H}' \leftarrow \mathbb{F}_2^{2t\log(m/t) \times n/2}$ and $\mathbf{T} \leftarrow \mathbb{F}_2^{n/2 \times n}$. Since $\widetilde{t} < n/2 < n$, by basic linear algebra, it follows that $\mathbf{H}$ is identically distributed as in Hyb_1.

  We now show that $\Pr[\mathsf{Hyb}_2 \text{ returns } 1] \leq \mathsf{negl}(\lambda)$ assuming DS-LPN holds. We do this by constructing an adversary $\mathcal{B}$ against DS-LPN from an adversary $\mathcal{A}$ against Hyb_2. $\mathcal{B}$ receives $(\mathbf{A}, \mathbf{b})$ where $\mathbf{A} = \mathbf{T} \cdot \mathbf{M}$ for $\mathbf{T} \leftarrow \mathbb{F}_2^{n/2 \times n}$, $\mathbf{M} \leftarrow \mathcal{M}_{\mathsf{sp}}$, and $\mathbf{b} \in \mathbb{F}_2^{1 \times m}$ is either uniformly random or is equal to $\mathbf{sA} + \mathbf{e}$ for a random $\mathbf{s} \leftarrow \mathbb{F}_2^{1 \times n/2}$ and $\mathbf{e} \leftarrow \mathsf{Ber}(\epsilon)^{1 \times m}$. $\mathcal{B}$ now samples $\mathbf{H}' \leftarrow \mathbb{F}_2^{(1-\rho)\widetilde{t} \times n/2}$, computes $\mathbf{A}' = \mathbf{H}' \cdot \mathbf{A}$, then runs $\mathcal{A}$ on $\mathbf{A}'$ to get $(\mathbf{x}_1, \mathbf{x}_2)$ from $\mathcal{A}$ and return $\mathsf{ans} := \mathbf{b} \cdot (\mathsf{spfy}_{m,t}(\mathbf{x}_1) - \mathsf{spfy}_{m,t}(\mathbf{x}_2))$. Here $\mathsf{ans} = 1$ indicates that $\mathcal{B}$ received random $\mathbf{b} \leftarrow \mathbb{F}_2^{1 \times m}$.

  We will analyze $\mathcal{B}$'s success probability. If $\mathbf{b}$ is uniformly random, then ans is also uniformly random whenever $\mathcal{A}$ wins (so that $\mathbf{x}_1 \neq \mathbf{x}_2$); thus, in this case we have

$$\Pr[\mathsf{ans} = 1 \mid \mathbf{b} \leftarrow \mathbb{F}_2^m] = \frac{1}{2} \cdot \Pr[\mathsf{Hyb}_2 \text{ returns } 1].$$

  If $\mathbf{b} = \mathbf{sA} + \mathbf{e}$, then $\mathsf{ans} = (\mathbf{sA} + \mathbf{e}) \cdot \mathbf{x}' = \mathbf{e} \cdot \mathbf{x}'$ whenever $\mathcal{A}$ wins in Hyb_2. Since $\mathbf{x}'$ is at most $2t$-sparse, by Lemma 3.2 we have

$$\begin{aligned} \Pr[\mathsf{ans} = 1 \mid \mathbf{b} = \mathbf{sA} + \mathbf{e}] &\leq \Pr[\mathbf{e} \cdot \mathbf{x}' = 1] \cdot \Pr[\mathsf{Hyb}_2 \text{ returns } 1] \\ &\leq \left(\frac{1 - (1-2\epsilon)^{2t}}{2}\right) \cdot \Pr[\mathsf{Hyb}_2 \text{ returns } 1] \\ &\leq \left(\frac{1}{2} - 2^{-8\epsilon t - 1}\right) \cdot \Pr[\mathsf{Hyb}_2 \text{ returns } 1] \\ &= \left(\frac{1}{2} - \frac{1}{\mathsf{poly}(\lambda)}\right) \cdot \Pr[\mathsf{Hyb}_2 \text{ returns } 1]. \end{aligned}$$

The last inequality is due to our choice of parameter $\epsilon = O(\log n/t)$, which implies $2^{-8\epsilon t-1} = 1/\mathsf{poly}(n) = 1/\mathsf{poly}(\lambda)$. Putting everything together, we have

$$\begin{aligned}\mathbf{Adv}^{\mathsf{DS\text{-}LPN}}(\mathcal{B}) &= |\Pr[\mathsf{ans} = 1 \mid \mathbf{b} = \mathbf{sA} + \mathbf{e}] - \Pr[\mathsf{ans} = 1 \mid \mathbf{b} \leftarrow \mathbb{F}_2^m]| \\ &\geq \frac{1}{\mathsf{poly}(\lambda)} \cdot \Pr[\mathsf{Hyb}_2 \text{ returns } 1].\end{aligned}$$

Therefore, if DS-LPN holds (for our parameters), then

$$\Pr[\mathsf{Hyb}_2 \text{ returns } 1] \leq \mathsf{poly}(\lambda) \cdot \mathbf{Adv}^{\mathsf{DS\text{-}LPN}}(\mathcal{B}) \leq \mathsf{negl}(\lambda).$$

□

6 Lossy Trapdoor Functions

In this section, we present our construction of lossy trapdoor functions (LTDFs) from Dense-Sparse LPN. In the full version, we will also present our direct construction of collision-resistant hash functions (CRHFs) from the same assumption.

6.1 Definition

We define all-injective-but-one lossy trapdoor functions, which include lossy trapdoor functions as a special case of having two branches.

Definition 6.1. *An* all-(injective)-but-one lossy trapdoor function $\mathsf{ABO\text{-}LTDF}$ *with input size $n = n(\lambda)$, output size $m = m(\lambda)$, residual leakage $r = r(\lambda)$, and number of branches $B(\lambda)$, consists of a PPT algorithm* Gen *and a tuple of deterministic functions (F, F^{-1}) with the following syntax:*

- $\mathsf{Gen}(1^\lambda, b^*) \to (\mathsf{fk}, \mathsf{td})$. *Given the security parameter 1^λ and a distinguished lossy branch $b^* \in [B(\lambda)]$, return a function key* fk *and a trapdoor* td.
- $F(\mathsf{fk}, b, x) \to y$. *Given the function key* fk*, a branch $b \in [B]$, and an input $x \in \mathbb{F}_2^n$, return an output $y \in \mathbb{F}_2^m$.*
- $F^{-1}(\mathsf{td}, b, y) \to x$. *Given the trapdoor* td*, a branch $b \in [B]$, and an output $y \in \mathbb{F}_2^m$, return a preimage $x \in \mathbb{F}_2^n$.*

and the following requirements:

- ***Branch Indistinguishability.*** *For any two different branches $b_0^* \neq b_1^* \in [B]$, the following two distributions are indistinguishable*

$$\left\{\mathsf{fk} \;\middle|\; (\mathsf{fk}, \mathsf{td}) \leftarrow \mathsf{Gen}(1^\lambda, b_0^*)\right\}_{\lambda\in\mathbb{N}} \approx_c \left\{\mathsf{fk} \;\middle|\; (\mathsf{fk}, \mathsf{td}) \leftarrow \mathsf{Gen}(1^\lambda, b_1^*)\right\}_{\lambda\in\mathbb{N}}.$$

- ***Correct Inversion for Injective Branch.*** *For any $\lambda \in \mathbb{N}$ and $b \neq b^* \in [B]$, we have*

$$\Pr\left[F^{-1}(\mathsf{td}, b, F(\mathsf{fk}, b, x)) = x \;\forall x \in \mathbb{F}_2^n \;\middle|\; (\mathsf{fk}, \mathsf{td}) \leftarrow \mathsf{Gen}(1^\lambda, b^*)\right] \geq 1 - \mathsf{negl}(\lambda).$$

- ***Lossyness for Lossy Branch.*** *For any $\lambda \in \mathbb{N}$, we have*

$$\Pr\left[|\{F(\mathsf{fk}, b^*, x) \mid x \in \mathbb{F}_2^n\}| \leq 2^r \;\middle|\; (\mathsf{fk}, \mathsf{td}) \leftarrow \mathsf{Gen}(1^\lambda, b^*)\right] \geq 1 - \mathsf{negl}(\lambda).$$

  We define the lossiness factor *to be $\Gamma = n/r$, meaning that in lossy mode, the output size is reduced by a factor of Γ compared to the input size.*

A lossy trapdoor function LTDF *is an* $\mathsf{ABO\text{-}LTDF}$ *with only two branches, i.e. $B(\lambda) = 2$ for all $\lambda \in \mathbb{N}$.*

Other extension of LTDFs. Over the years, several variants of LTDFs have been proposed with a goal toward increased functionality and more diverse applications. These include all-but-N LTDFs [HLOV11] (which can be built in a black-box way from LTDFs), all-but-many [Hof12], and cumulative-all-lossy-but-one [CPW20] lossy trapdoor functions. However, post-quantum constructions of the latter two variants [BL17, LSSS17, LNP22] rely on lattice techniques such as preimage sampling [GPV08, MP12] and GSW-style homomorphic evaluation [GSW13]. Since we do not know analogues of these techniques for code-based cryptography, we leave as future work the task of constructing these more advanced variants of LTDFs from code-based assumptions.

6.2 Construction

We give our construction of ABO-LTDF in Figure 3. In the construction, we use the following family of matrices $\{\mathbf{H}_\tau\}_{\tau\in\mathbb{F}_{2^n}}$ for any $n \in \mathbb{N}$, where $\mathbf{H}_\tau \in \mathbb{F}_2^{n\times n}$ is the matrix corresponding to multiplication by τ in the field $\mathbb{F}_{2^n}$ (which is isomorphic as vector spaces to $\mathbb{F}_2^n$). It follows that $\mathbf{H}_\tau - \mathbf{H}_{\tau'} = \mathbf{H}_{\tau-\tau'}$ is invertible for all $\tau \neq \tau'$. Such a family has been used in previous works [ABB10, KMP14].

To analyze the security of our construction, we begin with the following result which bounds the noise growth.

Lemma 6.1. *Consider the parameters $n, m, t, B, \epsilon, \ell$ as in Figure 3. Let $\mathbf{E} \leftarrow \mathsf{Ber}(\mathbb{F}_2, \epsilon)^{\ell\times m}$. Then except with $\mathsf{negl}(\lambda)$ probability over the choice of $\mathbf{E}$, the vector $\mathbf{E}\cdot\mathsf{spfy}_{m,t}(\mathbf{x}) \in \mathbb{F}_2^\ell$ is at most $\gamma\ell$-sparse for all $\mathbf{x} \in \mathbb{F}_2^{t\log(m/t)}$.*

Proof. By union bound, it suffices to show that for any fixed $\mathbf{x} \in \mathbb{F}_2^{t\log(m/t)}$, letting $\widetilde{\mathbf{x}} = \mathsf{spfy}_{m,t}(\mathbf{x})$, we have:

$$\Pr\left[\mathsf{wt}(\mathbf{E}\cdot\widetilde{\mathbf{x}}) > \gamma\ell \;\middle|\; \mathbf{E} \leftarrow \mathsf{Ber}(\mathbb{F}_2,\epsilon)^{\ell\times m}\right] \le 2^{-t\log(m/t)}\cdot\mathsf{negl}(\lambda).$$

Let $\mathbf{e} = \mathbf{E}\cdot\widetilde{\mathbf{x}}$. We can see that for each $j \in [\ell]$, the entry $(\mathbf{E}\cdot\widetilde{\mathbf{x}})_j = \langle\mathbf{E}_j, \widetilde{\mathbf{x}}\rangle$, where $\mathbf{E}_j$ is the j^{th} row of $\mathbf{E}$, is drawn from the Bernoulli distribution with noise

$$\epsilon' = \frac{1-(1-2\epsilon)^t}{2} < \epsilon t = \frac{\gamma}{\alpha+1}, \qquad \text{by Lemma 3.1 and the choice of } \epsilon.$$

We now apply Chernoff bound (Lemma 3.3) to get

$$\Pr\left[\mathsf{wt}(\mathbf{E}\cdot\widetilde{\mathbf{x}}) > \gamma\ell \;\middle|\; \mathbf{E} \leftarrow \mathsf{Ber}(\mathbb{F}_2,\epsilon)^{\ell\times m}\right] \le e^{-2\alpha^2\frac{\gamma}{\alpha+1}\ell} = e^{-2\frac{\alpha^2}{\alpha+1}\frac{\gamma}{\rho_{\mathcal{C}}}t\log(m/t)} < 2^{-2t\log(m/t)}.$$

In the above, the last inequality follows from our choice of α. Therefore, $\mathbf{E}\cdot\widetilde{\mathbf{x}}$ has Hamming weight at most $\gamma\ell$ for all $\mathbf{x} \in \mathbb{F}_2^{t\log(m/t)}$ with probability at least

$$1 - 2^{t\log(m/t)}\cdot 2^{-2t\log(m/t)} = 1 - 2^{-t\log(m/t)} = 1 - \mathsf{negl}(\lambda).$$

□

Theorem 6.1. *Assume the $(n, m, k, \mathcal{M}_{\mathsf{sp}}, \epsilon)$-Dense-Sparse LPN assumption holds, where $n, m, k, \mathcal{M}_{\mathsf{sp}}, \epsilon$ are chosen as in Figure 3. Then the construction in Figure 3 is an* ABO-LTDF *with input length $t\log(m/t)$, branches indexed by $\mathbb{F}_2^{t\log(m/t)}$, and lossiness factor $\Gamma > 1$.*

ABO-LTDFs from Dense-Sparse LPN

Parameters. Let $k \in \mathbb{N}, k \geq 3$ be a constant, $\Gamma > 1$ be any desired lossiness factor.

- Let $D > \Gamma$ be the compression factor, and consider any $\delta \in (\delta_{(4.1)}(k, D), 1)$.
- Let $n = \mathsf{poly}(\lambda)$, $m = m_{(4.1)}(n, k, D, \delta)$, $t = n^{\delta}$, and consider a good distribution $\mathcal{M}_{\mathsf{sp}}$ over $\mathsf{SpMat}(n, m, k)$.
- Let $\mathcal{C} = \{\mathbf{C}_\kappa\}_{\kappa \in \mathbb{N}}$ be an explicit family of asymptotically-good linear codes, where each $\mathbf{C}_\kappa$ has block length $L(\kappa)$, constant rate $\rho_{\mathcal{C}}$, and admits an efficient algorithm Decode that can correct up to $\delta_{\mathcal{C}} L$ errors (which exists by Lemma 3.7).
- Let $D' = \frac{\Gamma D}{\Gamma - D}$ so that $\frac{1}{D} + \frac{1}{D'} = \frac{1}{\Gamma}$, and $\gamma = \min\left(\delta_{\mathcal{C}}, H^{-1}\left(\frac{\rho}{D'}\right)\right)$. Let $\alpha > 0$ be a constant such that $\frac{\alpha^2}{\alpha+1} > \frac{\rho_{\mathcal{C}}}{\gamma}$.
- Let $\ell = \frac{1}{\rho_{\mathcal{C}}} \cdot t \log(\frac{m}{t})$ be the block length,[a] and $\epsilon = \frac{\gamma}{(\alpha+1)t}$ be the noise rate. We will abbreviate $\mathbf{C} = \mathbf{C}_\ell \in \mathbb{F}_2^{t \log(m/t) \times \ell}$.

Construction.

- $\mathsf{Gen}(1^\lambda, \tau^*) \to (\mathsf{fk}, \mathsf{td})$. Given lossy branch $\tau^* \in \mathbb{F}_{2^{t \log(m/t)}}$, sample $\mathbf{M} \leftarrow \mathcal{M}_{\mathsf{sp}}$, $\mathbf{T} \leftarrow \mathbb{F}_2^{n/2 \times n}$, $\mathbf{S} \leftarrow \mathbb{F}_2^{\ell \times n/2}$, and $\mathbf{E} \leftarrow \mathsf{Ber}(\mathbb{F}_2, \epsilon)^{\ell \times m}$.

  Let $\mathbf{A} = \mathbf{T} \cdot \mathbf{M} \in \mathbb{F}_2^{n/2 \times m}$, and $\mathbf{B} = \mathbf{S} \cdot \mathbf{A} + \mathbf{E} + \mathbf{C}^\mathsf{T} \cdot \mathbf{H}_{\tau^*} \cdot \mathbf{G}_{m,t}^\mathsf{T} \in \mathbb{F}_2^{\ell \times m}$.

  Return $\mathsf{fk} = (\mathbf{A}, \mathbf{B})$ and $\mathsf{td} = (\mathbf{S}, \tau^*)$.
- $F(\mathsf{fk}, \tau, \mathbf{x}) \to \mathbf{y}$. Given $\tau \in \mathbb{F}_{2^{t \log(m/t)}}$ and $\mathbf{x} \in \mathbb{F}_2^{t \log(m/t)}$, compute $\widetilde{\mathbf{x}} = \mathsf{spfy}_{m,t}(\mathbf{x})$ and $\mathbf{B}_\tau = \mathbf{B} - \mathbf{C}^\mathsf{T} \cdot \mathbf{H}_\tau \cdot \mathbf{G}_{m,t}^\mathsf{T}$.

  Return $\mathbf{y} = (\mathbf{A} \cdot \widetilde{\mathbf{x}}, \mathbf{B}_\tau \cdot \widetilde{\mathbf{x}}) \in \mathbb{F}_2^{n/2+\ell}$.
- $F^{-1}(\mathsf{td}, \tau, \mathbf{y}) \to \mathbf{x}$. Parse $\mathbf{y} = (\mathbf{y}_1 \in \mathbb{F}_2^{n/2}, \mathbf{y}_2 \in \mathbb{F}_2^\ell)$ and compute $\mathbf{y}' = \mathbf{y}_2 - \mathbf{S} \cdot \mathbf{y}_1$.

  Return $\mathbf{x} \leftarrow (\mathbf{H}_{\tau^* - \tau})^{-1} \cdot \mathsf{Decode}(\mathbf{y}')$.

[a] Without loss of generality, we may assume $\ell = L(\kappa)$ for some $\kappa \in \mathbb{N}$. Otherwise, letting $L(\kappa)$ be the nearest block size larger than ℓ, we may scale n (and thus ℓ) by an appropriate polynomial amount so that $\ell = L(\kappa)$.

Figure 3: ABO-LTDF construction

Proof. We argue each property separately as follows.

Mode indistinguishability. This is clear from the Dense-Sparse LPN assumption, since in both cases, $\mathsf{fk} = (\mathbf{A}, \mathbf{B})$ is indistinguishable from $(\mathbf{A}, \mathbf{U})$ for a random $\mathbf{U} \leftarrow \mathbb{F}_2^{\ell \times m}$.

Trapdoor inversion. If $\mathsf{Gen}(1^\lambda, \mathsf{inj}) \to (\mathsf{fk} = (\mathbf{A}, \mathbf{B}), \mathsf{td} = (\mathbf{S}, \tau^*))$ and $F(\mathsf{fk}, \tau, \mathbf{x}) = (\mathbf{y}_1, \mathbf{y}_2)$ for some $\mathbf{x} \in \mathbb{F}_2^{t \log(m/t)}$, letting $\widetilde{\mathbf{x}} = \mathsf{spfy}_{m,t}(\mathbf{x})$, we have that

$$\begin{aligned} \mathbf{y}_1 = \mathbf{A} \cdot \widetilde{\mathbf{x}}, \quad \mathbf{y}_2 &= (\mathbf{B} - \mathbf{C}^\mathsf{T} \cdot \mathbf{H}_\tau^\mathsf{T} \cdot \mathbf{G}_{m,t}^\mathsf{T}) \cdot \widetilde{\mathbf{x}} \\ &= \mathbf{S} \cdot \mathbf{y}_1 + \mathbf{E} \cdot \widetilde{\mathbf{x}} + \mathbf{C}^\mathsf{T} \cdot \mathbf{H}_{\tau^* - \tau} \cdot \mathbf{x}. \end{aligned}$$

The last equality follows from Equation (8), namely that $\mathbf{G}_{m,t}^{\mathsf{T}} \cdot \mathsf{spfy}_{m,t}(\mathbf{x}) = \mathbf{x}$. Hence for all $\tau \neq \tau^*$, with all but $\mathsf{negl}(\lambda)$ probability, we have:

$$\begin{aligned} F^{-1}(\mathsf{td}, (\mathbf{y}_1, \mathbf{y}_2)) &= (\mathbf{H}_{\tau^* - \tau})^{-1} \cdot \mathsf{Decode}\left(\mathbf{y}_2 - \mathbf{S} \cdot \mathbf{y}_1\right) \\ &= (\mathbf{H}_{\tau^* - \tau})^{-1} \cdot \mathsf{Decode}\left(\mathbf{C}^{\mathsf{T}} \cdot \mathbf{H}_{\tau^* - \tau} \cdot \mathbf{x} + \mathbf{E} \cdot \mathsf{spfy}_{m,t}(\mathbf{x})\right) \\ &= (\mathbf{H}_{\tau^* - \tau})^{-1} \cdot (\mathbf{H}_{\tau^* - \tau} \cdot \mathbf{x}) \qquad \text{(holds with } 1 - \mathsf{negl}(\lambda) \text{ probability)} \\ &= \mathbf{x}. \end{aligned}$$

Indeed, the third equality holds with probability $1 - \mathsf{negl}(\lambda)$ because of the following. First, the fact that Decode can efficiently decode from any $\gamma\ell$ errors, i.e. $\mathsf{Decode}(\mathbf{C}^{\mathsf{T}} \cdot \mathbf{x} + \mathbf{e}) = \mathbf{x}$ for any $\mathbf{e} \in \mathcal{B}_{\leq}(\ell, \gamma\ell)$. Second, from Lemma 6.1 and the choice of γ, we know that $\mathsf{wt}(\mathbf{E} \cdot \mathsf{spfy}_{m,t}(\mathbf{x})) \leq \gamma\ell \leq \delta_{\mathcal{C}}\ell$ for any $\mathbf{x} \in \mathbb{F}_2^{t \log(m/t)}$, which holds with all but $\mathsf{negl}(\lambda)$ probability.

Lossyness. For the lossy branch $\tau = \tau^*$, we will count the number of attainable values $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2) \in \mathbb{F}_2^{n/2+\ell}$ in the range of F. Denote $\mathcal{X} = \mathbb{F}_2^{t \log(m/t)}$ to be the domain, and $\mathcal{Y} = \{(\mathbf{y}_1, \mathbf{y}_2)\} \subset \mathbb{F}_2^{n/2+\ell}$ to be the range.

The first part $\mathbf{y}_1 := \mathbf{A} \cdot \widetilde{\mathbf{x}} = \mathbf{T} \cdot (\mathbf{M} \cdot \widetilde{\mathbf{x}})$, is determined by the value $\mathbf{M} \cdot \widetilde{\mathbf{x}} \in \mathcal{B}(n, \leq kt)$, hence has cardinality at most $|\mathcal{B}_{\leq}(n, kt)|$. By our choice of parameters for n, m, t so that Lemma 4.1 is satisfied with compression factor D, it follows that the set $\mathcal{Y}_1$ of possible values of $\mathbf{y}_1$ satisfies $|\mathcal{Y}_1| < |\mathcal{X}|^{1/D} = 2^{t \log(m/t)/D}$.

Next, for a fixed $\mathbf{y}_1$, using Lemma 6.1, we have that

$$\mathbf{y}_2 = \mathbf{B}_\tau \cdot \widetilde{\mathbf{x}} = \mathbf{S} \cdot \mathbf{y}_1 + \mathbf{E} \cdot \widetilde{\mathbf{x}}$$

is in a Hamming ball of radius $\gamma\ell$ around $\mathbf{y}_1$ except with $\mathsf{negl}(\lambda)$ probability. Thus we have that

$$\begin{aligned} |\mathcal{Y}| &\leq |\mathcal{Y}_1| \cdot |\mathcal{B}_{\leq}(\ell, \gamma\ell)| \\ &\leq 2^{t \log(m/t)/D} \cdot 2^{H(\gamma)\ell} \\ &\leq \left(2^{t \log(m/t)}\right)^{1/D + 1/D'} \\ &= |\mathcal{X}|^{1/\Gamma}, \end{aligned}$$

and thus the lossiness parameter is at least Γ. Here the second inequality is due to upper bounds on $|\mathcal{Y}_1|$ and $|\mathcal{B}_{\leq}(\ell, \gamma\ell)|$, the third inequality is due to our choice of $\gamma \leq H^{-1}(\rho/D')$, which implies that $H(\gamma)\ell = \frac{H(\gamma)}{\rho} \cdot t \log(m/t) \leq \frac{1}{D'} \cdot t \log(m/t)$, and the final equality is because $\frac{1}{D} + \frac{1}{D'} = \frac{1}{\Gamma}$. □

7 Cryptanalysis of Dense-Sparse LPN

Recent works, in the context of generating correlated pseudorandomness for MPC applications [BCGI18, BCG+19, BCG+20a], have proposed various novel variants of LPN with different matrix distributions [BCG+20a, CRR21, BCG+22, CD23b, RRT23, BCCD23]. These recent progress also came with a *systematization* of known attacks on LPN-style assumption. Namely, prior works observed that most attacks (Information-Set Decoding [Pra62], BKW [BKW00], Gaussian Elimination [EKM17], Statistical Decoding [Jab01], etc.) on LPN-style assumptions can be captured by a unified framework for the security of LPN variants, namely the *linear test framework*, which we will also use to analyze the security of our assumption. The linear test framework has also been used to extensively cryptanalyze Goldreich's PRGs [Gol00, CM01, MST03, CEMT09, BQ09,

ABW10, ABR12, BQ12, App12, App13, OW14, AL16, KMOW17, CDM+18, AK19], which is closely related to Sparse LPN.

In a linear test, an adversary takes as input a matrix $\mathbf{A} \in \mathbb{Z}_2^{n \times m}$ and outputs a vector $\mathbf{v} \in \mathbb{Z}_2^{m \times 1}$ such that $(\mathbf{sA} + \mathbf{e})\,\mathbf{v}$ is a biased random variable with an inverse polynomial bias towards 0. Assuming $\mathbf{e}$ is chosen from Bernoulli distribution with probability ϵ, a successful attacker must output $\mathbf{v}$ with hamming weight $O(\frac{\log n}{\epsilon})$, or else the bias is negligible.

Definition 7.1 (Security against Linear Test, adapted from [CRR21])**.** *Let $n \in \mathbb{N}$ be the dimension, $m = m(n)$ be the number of samples, and $q = q(n)$ be a prime power. Given an efficiently sampleable distribution $\mathcal{M} = \mathcal{M}(n, m, \mathbb{F}_q)$ over matrices in $\mathbb{F}_q^{n \times m}$ and noise probability $\epsilon = \epsilon(n) \in (0, 1)$, we say that the $(\mathcal{M}, \epsilon)$-*LPN *assumption is $(T(n), \alpha(n), \delta(n))$-*computationally secure against linear tests *if for any adversary $\mathcal{A}$ running in time at most $T(n)$, it holds that*

$$\Pr\left[\mathsf{bias}_{\mathbf{v}}(\mathcal{D}_{\mathbf{A}}) \geq \alpha \;\middle|\; \begin{array}{l} \mathbf{A} \leftarrow \mathcal{M} \\ \mathbf{v} \leftarrow \mathcal{A}(\mathbf{A}) \end{array}\right] \leq \delta,$$

where $\mathcal{D}_{\mathbf{A}} = \left\{\mathbf{sA} + \mathbf{e} \mid \mathbf{s} \leftarrow \mathbb{F}_q^{1 \times n}, \mathbf{e} \leftarrow \mathsf{Ber}(\mathbb{F}_q, \epsilon)^{1 \times m}\right\}$. (Recall that bias is defined in Definition 3.1.)

In other words, the Linear Test Hypothesis for the $(\mathcal{M}, \epsilon)$-LPN assumption states that $(\mathcal{M}, \epsilon)$-LPN is secure if it is computationally difficult to find $O(\frac{\log n}{\epsilon})$-sparse vectors in the kernel of $\mathbf{A} \leftarrow \mathcal{M}$. All known counter-examples to this hypothesis are for distributions $\mathcal{M}$ of algebraically structured matrices (see [BCG+20b] for a detailed discussion). In contrast, our Dense-Sparse matrix distribution $\mathbf{A} = \mathbf{TM}$ has no apparent algebraic structure, and thus it is reasonable to conjecture that Dense-Sparse LPN is secure assuming it is secure against linear tests. Furthermore, if finding $O(\frac{\log n}{\epsilon})$-sparse vectors in the kernel is difficult for subexponential time adversaries, then the corresponding LPN assumption is also subexponentially secure.

Assuming the Linear Test Hypothesis, we now examine the hardness of finding sparse vectors in the kernel of $\mathbf{A} = \mathbf{TM}$. Such vectors of sparsity n^δ for any $\delta \in (0, 1)$ actually come from the kernel of $\mathbf{M}$, with all but negligible probability. This is because $\mathbf{T}$ is a random binary matrix of dimension $\frac{n}{2} \times n$, and so typically only has vectors in the kernel that are at least $\Omega(n/\log n)$-sparse (see [CRR21] for a detailed calculation). Therefore, we will focus our attention on the size of short vectors in the kernel of the k-sparse matrix $\mathbf{M}$, and also on the overall security of Sparse LPN.

7.1 Security of Sparse LPN

Sparse LPN is a relatively well-understood assumption in the domain of refutations for random constraint satisfaction. When one samples a matrix $\mathbf{M}$ for Sparse LPN with sparsity parameter k, dimension n, and sample complexity $\Omega\left(n^{1+(\frac{k}{2}-1)(1-\delta)}\right)$, it can be proven that the dual distance of $\mathbf{M}$, chosen randomly from $\mathsf{SpMat}(n, m, k, \mathbb{F}_2)$ (the set of k-sparse matrices in $\mathbb{F}_2^{n \times m}$), is $t = \Omega(n^\delta)$ with all but inverse polynomial probability.

Lemma 7.1 (Folklore, see [DIJL23])**.** *Given any $k = k(n) \geq 3$, $0 < \delta < 1$, and $m = \Omega\left(n^{1+(\frac{k}{2}-1)(1-\delta)}\right)$, there exists a constant $c > 0$ such that the following holds for large enough n:*

$$\Pr\left[\mathsf{dd}(\mathbf{M}) \leq c \cdot n^\delta \;\middle|\; \mathbf{M} \leftarrow \mathsf{SpMat}(n, m, k, \mathbb{F}_2)\right] = O\left(\left(\frac{k}{n^\delta}\right)^{k-2}\right).$$

Note that the dual distance of $t = \Omega(n^\delta)$ is also known in the stronger case of *worst-case* k-sparse $\mathbf{M}$, whenever $m = \widetilde{\Omega}\left(n^{1+(\frac{k}{2}-1)(1-\delta)}\right)$ [GKM22, HKM23].

One could then consider Sparse LPN in two regimes. In the first regime, the error probability is $\epsilon = \omega(\frac{\log n}{t})$. In this regime, conditioned on the event that there simply *does not exist* vectors of sparsity $O(\frac{\log n}{\epsilon})$ in the kernel of $\mathbf{M}$, security against linear tests holds. Sparse LPN with this regime of error (even constant error probability) has found prior applications [IKOS08, ADI+17, AK23, BCG+23, DIJL23].

Nevertheless, there are also applications of Sparse LPN where sparse vectors (in the kernel of $\mathbf{M}$) do exist, but seem computationally hard to find. One such example is the public-key encryption scheme of Applebaum, Barak and Wigderson [ABW10] uses an inverse polynomial error probability $\epsilon = O(\frac{\log n}{t})$. In this case, certainly there exist lots of t-sparse vectors in the kernel of $\mathbf{M}$. To this date, we do not know any procedure that can efficiently find (even in subexponential time 2^{t^ρ} for any $0 < \rho < 1$) such t-sparse vectors in the kernel of $\mathbf{M}$. Therefore, the linear test framework correctly predicts the (subexponential) security of this assumption.

Our applications could have been based on Sparse LPN in its compression regime. This holds at a significantly lower error probability $\epsilon_{\mathsf{cps}} = o(\frac{1}{t_{\mathsf{cps}}})$ where $t_{\mathsf{cps}} \sim \left(\frac{n^k}{m}\right)^{\frac{1}{k-1}}$, which is polynomially larger than the dual distance $t \sim \left(\frac{n^{k/2}}{m}\right)^{\frac{1}{k/2-1}}$. We in fact show that it is *easy* to find t_{cps}-sparse solutions to $\mathbf{M}\,\mathbf{x}$ for randomly chosen $\mathbf{M}$.

Sparse LPN is Broken in its Compression Regime. Following our exposition in Section 2, we give a simple attack on Sparse LPN in the parameter regime needed for achieving compression (and hence lossy trapdoor functions).

Theorem 7.1. *Consider Sparse LPN with sparsity k, dimension n and sample complexity m in the regime in Lemma 4.1, with error rate $\epsilon = O(1/t)$. Assuming the matrix $\mathbf{M}$ is chosen by sampling distinct k-sparse random columns, e.g. $\mathbf{M} \leftarrow \mathsf{SpMat}(n, m, k, \mathbb{F}_2)$, Sparse LPN with these parameters can be broken in polynomial time.*

The attack is as follows. Given samples $(\mathbf{M}, \mathbf{b})$ with $\mathbf{b}$ either from the Sparse LPN distribution, or the random distribution, we do the following:

1. Pick a random subset $S \subset [n]$ of size t. Initialize $T = \emptyset$.
2. For each column $j \in [m]$ of $\mathbf{M}$, if all k of the non-zero entries lie inside S, add j to T.
3. If $|T| > t$, find a linear dependency (expressed as a vector $\mathbf{x}$) between the columns in T.
4. Compute $d = \langle \mathbf{b}, \mathbf{x} \rangle$. If $d = 0$, return "Sparse LPN", else return "random".

Lemma 7.2. *This attack succeeds with high probability $1 - o(1)$ whenever $m > c \cdot m_{(4.1)}(k, D)$ for a sufficiently large constant c and any compression factor $D > 1$.*

Proof. For any fixed choice of $S \subset [n]$, we can compute the expectation of the number of columns found in step 2 as follows.

$$\mathbb{E}[|T|] = m \cdot \Pr[\mathsf{Supp}(\mathbf{a}) \subset S \mid \mathbf{a} \leftarrow \mathsf{SpMat}(n, 1, k)] = m \cdot \frac{\binom{t}{k}}{\binom{n}{k}} \approx m \cdot \left(\frac{t}{n}\right)^k > t.$$

The last inequality is due to the choice $m > m_{(4.1)}(k, 1)$.

□

7.2 Dual-Distance of Dense-Sparse LPN

We now examine the dual-distance of the matrices arising in our Dense-Sparse LPN assumption, and the computational hardness of finding sparse vectors $\boldsymbol{x}$ that are in the kernel of those matrices. Since our matrices are of the form $\mathbf{A} = \mathbf{TM}$ where $\mathbf{M} \in \mathbb{Z}_2^{n \times m}$ is a k-sparse matrix, with high probability the dual distance of $\mathbf{M}$ (and hence of $\mathbf{A}$) is around $t = O(n^\delta)$ where $m = n^{1+(\frac{k}{2}-1)(1-\delta)}$ where as the error probability $\epsilon < \frac{1}{t_{\mathsf{cps}}}$, where $t_{\mathsf{cps}} \sim \left(\frac{n^k}{m}\right)^{\frac{1}{k-1}}$ is the threshold for compression.

Sparse Vector in the Kernel of TM. As described above, if we can efficiently find $t = O(n^\delta)$ sparse vectors $\boldsymbol{x} \in \mathbb{Z}_2^{m \times 1}$ in the kernel of $\mathbf{TM}$, that is enough to break our assumption with any error probability $O(\frac{\log n}{t})$. In our specific setting, we work with an error probability of about $\frac{1}{t_{\mathsf{cps}}}$ where $t_{\mathsf{cps}} \sim \left(\frac{n^k}{m}\right)^{\frac{1}{k-1}} \approx \frac{n}{m^{1/k}}$ (for large enough constant k). Setting $m = n^{1+(k/2-1)(1-\delta)}$, this turns out to be roughly $t_{\mathsf{cps}} \approx n^{\frac{1}{2}+\delta-\frac{1}{k}}$. On the other hand, much sparser vectors ($t = O(n^\delta)$ sparse) in the kernel exist. Thus, if one can find kernel vectors of this sparsity, it could break our assumption. However, for arbitrary dense matrices we don't know a better way to find such n^δ-sparse vectors significantly better than naive search, which is known to be subexponentially hard (runs in time $2^{\widetilde{O}(n^\delta)}$) even given $\mathbf{M}$ in the clear (this very assumption was made by [ABW10]).

It is also true that one could also break our assumption by finding vectors in the kernel with much higher sparsity $t_{\mathsf{cps}} \approx n^{\frac{1}{2}+\delta-\frac{1}{k}}$. In fact, we showed how to do find such vectors in Theorem 7.1, but our attack crucially relies on seeing the *sparsity pattern* of the matrix $\mathbf{M}$. This pattern is no longer apparent when we only give out $\mathbf{A} = \mathbf{TM}$ with $\mathbf{T}$ being a random (dense) matrix, so the attack in Theorem 7.1 no longer applies.

7.2.1 Searching for the Inner Matrix T

Next, we examine the possibility of learning $\mathbf{T} \in \mathbb{Z}_2^{\alpha n \times n}$ from the matrix $\mathbf{TM}$ and then finding t_{cps}-sparse vectors in the kernel of $\mathbf{M}$. From a simple entropy calculation, the decomposition $\mathbf{TM}$ is unique with high probability (up to left multiplying by an invertible matrix, and right multiplying by a permutation matrix). The question is whether one can efficiently recover the decomposition. We show that this is possible if $\alpha \geq 1$, and provide evidence that it is infeasible when $\alpha < 1$. This justifies our choice of the parameter α for Dense-Sparse LPN.

Attack when $\alpha \geq 1$. We can generalize our previous attack to "unmask" $\mathbf{T}$ in this setting by searching for its left inverse $\mathbf{Z} \in \mathbb{Z}_2^{n \times \alpha n}$. In other words, given the Dense-Sparse matrix $\mathbf{A} = \mathbf{T} \cdot \mathbf{M}$, we want to find $\mathbf{Z} \in \mathbb{F}_2^{n \times \alpha n}$ such that $\mathbf{Z} \cdot \mathbf{A} = \mathbf{M}$ is a sparse matrix. We will find $\mathbf{Z}$ row-by-row; let a candidate row for $\mathbf{Z}$ be $\mathbf{z}$. Observe that $\mathbf{zA}$ must be a sparse vector that corresponds to a row of $\mathbf{M}$. Since each column of $\mathbf{M}$ has some constant number k of non-zero entries, each row of $\mathbf{M}$ has density of non-zero entries roughly $\approx k/n$ as well. Thus, one could randomly sample $2n$ indices of $\mathbf{zA}$, assume that they are zero, and solve for $\mathbf{z}$ using these equations. With probability roughly $(1 - \frac{k}{n})^{2n} = \Omega(1)$, the assumption will be correct, and we will be able to solve for $\mathbf{z}$ with high probability ($2n$ equations to find $\mathbf{z} \in \mathbb{Z}_2^n$; one should be able to find n independent vectors from the corresponding columns of $\mathbf{A}$).

Attack fails when $\alpha < 1$. In this setting, $\mathbf{T}$ no longer has a left inverse. One natural way to fix the above attack would be to find an inverse $\mathbf{Z}$ for a square sub-matrix of $\mathbf{T}$. For example, we can

hope to find $\mathbf{Z} \in \mathbb{Z}_2^{\alpha n \times \alpha n}$ that is the inverse of the submatrix formed by the first $\alpha \cdot n$ sub-columns of $\mathbf{T}$, so that $\mathbf{ZTM} = [\mathbf{I}_{\alpha n} \| \mathbf{T}'] \cdot \mathbf{M}$. Note that $\mathbf{T}' \in \mathbb{F}_2^{\alpha n \times (1-\alpha)n}$ is randomly distributed.

Now, if one examines $[\mathbf{I}_{\alpha n} \| \mathbf{T}'] \cdot \mathbf{M}$ the following holds. For any i-th column $\mathbf{m}_i$ of $\mathbf{M}$, if $\mathbf{m}_i$ is supported over the first $\alpha \cdot n$ variables, then the same i-th column in $[\mathbf{I}_{\alpha n} \| \mathbf{T}'] \cdot \mathbf{M}$ is precisely $\mathbf{m}_i$, and hence remains k-sparse. Otherwise, if $\mathbf{m}_i$ has a non-zero entry in the rest of the positions, then the randomness of $\mathbf{T}'$ would make the resulting column in $\mathbf{ZTM}$ random as well.

Let's calculate when the first event happens, so that we have hope of unmasking a column of $\mathbf{M}$. Since there is a constant chance that when choosing a column of $\mathbf{M}$, that column is supported inside the first $\alpha \cdot n$ set of variables, the product $[\mathbf{I}_{\alpha n} \| \mathbf{T}'] \cdot \mathbf{M}$ will be so that each row has a constant fraction of non-zero element. As a consequence, to solve for $\mathbf{Z}$ we might have to rely on an LPN solver that works with constant probability noise which might be hard to do. In general, this attack should take (near-)exponential time as long $\mathbf{T}$ is randomly chosen from $\mathbb{Z}_2^{\alpha n \times n}$ for any $\alpha > 0$.

Algebraic methods for recovering $\mathbf{T}$. One could ask if Gröbner Basis style algorithms can learn $\mathbf{T}$, similar to the attack by Arora-Ge [AG11] on LPN variants with *regular* noise (instead of Bernoulli-distributed noise). The problem with this approach is that it is unclear how to set up algebraic constraints to enforce the constraint that a vector of variables is k-sparse. Such methods have been tried recently, and a recent paper [LSS23] explains the difficulties encountered by this approach.

7.3 Concluding the State of Attacks

From the above discussions, we conclude with the following known attacks on Dense-Sparse LPN:

- The first class are generic attacks on the LPN assumption, which are oblivious to the matrix. These attacks take $2^{\widetilde{O}(n\epsilon)}$ time, which is the case for, e.g., information-set decoding attacks that simply guesses n noiseless equations and solve for the secret.
- The second attack is to search for a $t = O(n^\delta)$-sparse vector $\mathbf{x} \in \mathbb{Z}_2^{m \times 1}$ such that $\mathbf{Mx} = \mathbf{0}$, where $\mathbf{M} \in \mathbb{Z}_2^{n \times m}$ is k-sparse and $m = n^{1+(k/2-1)(1-\delta)}$. This will allow us to solve Dense-Sparse LPN with error probability $O\left(\frac{\log n}{t}\right)$. Crucially, since there is no known algorithm with asymptotic better than just brute-force search, this attack would take time $2^{\widetilde{O}(n^\rho)}$.
- Finally, one can attempt to recover the trapdoor matrices $\mathbf{T}, \mathbf{M}$ from $\mathbf{A} = \mathbf{TM}$. This will allow breaking Dense-Sparse LPN with lower error rate (in the regime enabling LTDFs as in Section 6), since using knowledge of $\mathbf{M}$ one can find a much less sparse vector $\mathbf{x}$ Lemma 7.2. As we surveyed in Section 7.2.1, natural approaches to this take $\exp(\widetilde{\Omega}(n))$ time.

More concisely, our conjectured security can be summarized as follows:

Conjecture 7.1. *Using the notation of Definition 4.5, $(n, m, k, \mathcal{M}_{\mathsf{sp}}, \epsilon)$-DS-LPN is*

$$\left(T(n) := \min\left(2^{\widetilde{O}(n^\delta)}, 2^{\widetilde{O}(n\cdot\epsilon)}\right),\ \delta(n) := \mathsf{negl}(n)\right)\text{- hard.}$$

In fact, our cryptanalysis along with the Linear Test hypothesis support the stronger assumption that Dense-Sparse matrices appear pseudorandom (to subexponential-time adversaries). This would imply Conjecture 7.1 via reducing to the conjectured security of (standard) LPN.

Conjecture 7.2. *Using the notation of Definition 4.5, $\mathcal{M}_{\mathsf{DS}}$ is $\left(\min\left(2^{\widetilde{O}(n^\delta)}, 2^{\widetilde{O}(n\cdot\epsilon)}\right),\ \mathsf{negl}(n)\right)$ computationally indistinguishable from $\{\mathbf{A} \mid \mathbf{A} \leftarrow \mathbb{F}_2^{\alpha n \times m}\}$.*

8 References

[ABB10] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 553–572. Springer, Berlin, Heidelberg, May / June 2010. 11, 23

[ABR12] Benny Applebaum, Andrej Bogdanov, and Alon Rosen. A dichotomy for local small-bias generators. In Ronald Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 600–617. Springer, Berlin, Heidelberg, March 2012. 3, 26

[ABW10] Benny Applebaum, Boaz Barak, and Avi Wigderson. Public-key cryptography from different assumptions. In Leonard J. Schulman, editor, *42nd ACM STOC*, pages 171–180. ACM Press, June 2010. 3, 4, 5, 12, 26, 27, 28

[ADI+17] Benny Applebaum, Ivan Damgård, Yuval Ishai, Michael Nielsen, and Lior Zichron. Secure arithmetic computation with constant computational overhead. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 223–254. Springer, Cham, August 2017. 27

[ADMP20] Navid Alamati, Luca De Feo, Hart Montgomery, and Sikhar Patranabis. Cryptographic group actions and applications. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part II*, volume 12492 of *LNCS*, pages 411–439. Springer, Cham, December 2020. 6

[AFS05] Daniel Augot, Matthieu Finiasz, and Nicolas Sendrier. A family of fast syndrome based cryptographic hash functions. In *Progress in Cryptology–Mycrypt 2005: First International Conference on Cryptology in Malaysia, Kuala Lumpur, Malaysia, September 28-30, 2005. Proceedings 1*, pages 64–83. Springer, 2005. 5

[AG11] Sanjeev Arora and Rong Ge. New algorithms for learning in presence of errors. In Luca Aceto, Monika Henzinger, and Jiri Sgall, editors, *ICALP 2011, Part I*, volume 6755 of *LNCS*, pages 403–415. Springer, Berlin, Heidelberg, July 2011. 29

[AHI+17] Benny Applebaum, Naama Haramaty, Yuval Ishai, Eyal Kushilevitz, and Vinod Vaikuntanathan. Low-complexity cryptographic hash functions. In Christos H. Papadimitriou, editor, *ITCS 2017*, volume 4266, pages 7:1–7:31, 67, January 2017. LIPIcs. 5

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *28th ACM STOC*, pages 99–108. ACM Press, May 1996. 5

[AK19] Benny Applebaum and Eliran Kachlon. Sampling graphs without forbidden subgraphs and unbalanced expanders with negligible error. In David Zuckerman, editor, *60th FOCS*, pages 171–179. IEEE Computer Society Press, November 2019. 3, 4, 12, 18, 26

[AK23] Benny Applebaum and Niv Konstantini. Actively secure arithmetic computation and VOLE with constant computational overhead. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part II*, volume 14005 of *LNCS*, pages 190–219. Springer, Cham, April 2023. 27

[AL16] Benny Applebaum and Shachar Lovett. Algebraic attacks against random local functions and their countermeasures. In Daniel Wichs and Yishay Mansour, editors, *48th ACM STOC*, pages 1087–1100. ACM Press, June 2016. 3, 26

[Ale03] Michael Alekhnovich. More on average case vs approximation complexity. In *44th FOCS*, pages 298–307. IEEE Computer Society Press, October 2003. 1, 3

[AMR22] Navid Alamati, Giulio Malavolta, and Ahmadreza Rahimi. Candidate trapdoor claw-free functions from group actions with applications to quantum protocols. In Eike Kiltz and Vinod Vaikuntanathan, editors, *TCC 2022, Part I*, volume 13747 of *LNCS*, pages 266–293. Springer, Cham, November 2022. 6

[App12] Benny Applebaum. Pseudorandom generators with long stretch and low locality from random local one-way functions. In Howard J. Karloff and Toniann Pitassi, editors, *44th ACM STOC*, pages 805–816. ACM Press, May 2012. 3, 26

[App13] Benny Applebaum. Pseudorandom generators with long stretch and low locality from random local one-way functions. *SIAM J. Comput.*, 42(5):2008–2037, 2013. 3, 26

[BBC08] Marco Baldi, Marco Bodrato, and Franco Chiaraluce. A new analysis of the McEliece cryptosystem based on QC-LDPC codes. In Rafail Ostrovsky, Roberto De Prisco, and Ivan Visconti, editors, *SCN 08*, volume 5229 of *LNCS*, pages 246–262. Springer, Berlin, Heidelberg, September 2008. 4

[BBN+09] Mihir Bellare, Zvika Brakerski, Moni Naor, Thomas Ristenpart, Gil Segev, Hovav Shacham, and Scott Yilek. Hedged public-key encryption: How to protect against bad randomness. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 232–249. Springer, Berlin, Heidelberg, December 2009. 5

[BC07] Marco Baldi and Franco Chiaraluce. Cryptanalysis of a new instance of mceliece cryptosystem based on QC-LDPC codes. In *IEEE International Symposium on Information Theory, ISIT 2007, Nice, France, June 24-29, 2007*, pages 2591–2595. IEEE, 2007. 4

[BCCD23] Maxime Bombar, Geoffroy Couteau, Alain Couvreur, and Clément Ducros. Correlated pseudorandomness from the hardness of quasi-abelian decoding. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part IV*, volume 14084 of *LNCS*, pages 567–601. Springer, Cham, August 2023. 5, 25

[BCD+16] Magali Bardet, Julia Chaulet, Vlad Dragoi, Ayoub Otmani, and Jean-Pierre Tillich. Cryptanalysis of the McEliece public key cryptosystem based on polar codes. In Tsuyoshi Takagi, editor, *Post-Quantum Cryptography - 7th International Workshop, PQCrypto 2016*, pages 118–143. Springer, Cham, 2016. 4

[BCG+19] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. Efficient pseudorandom correlation generators: Silent OT extension and more. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 489–518. Springer, Cham, August 2019. 5, 25

[BCG+20a] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. Correlated pseudorandom functions from variable-density LPN. In *61st FOCS*, pages 1069–1080. IEEE Computer Society Press, November 2020. 5, 25

[BCG+20b] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. Efficient pseudorandom correlation generators from ring-LPN. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 387–416. Springer, Cham, August 2020. 5, 26

[BCG+22] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, Nicolas Resch, and Peter Scholl. Correlated pseudorandomness from expand-accumulate codes. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 603–633. Springer, Cham, August 2022. 5, 25

[BCG+23] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, Nicolas Resch, and Peter Scholl. Oblivious transfer with constant computational overhead. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part I*, volume 14004 of *LNCS*, pages 271–302. Springer, Cham, April 2023. 27

[BCGI18] Elette Boyle, Geoffroy Couteau, Niv Gilboa, and Yuval Ishai. Compressing vector OLE. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018*, pages 896–912. ACM Press, October 2018. 5, 25

[BCM+18] Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh V. Vazirani, and Thomas Vidick. A cryptographic test of quantumness and certifiable randomness from a single quantum device. In Mikkel Thorup, editor, *59th FOCS*, pages 320–331. IEEE Computer Society Press, October 2018. 1

[BF22] Nir Bitansky and Sapir Freizeit. Statistically sender-private OT from LPN and derandomization. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part III*, volume 13509 of *LNCS*, pages 625–653. Springer, Cham, August 2022. 2, 6

[BFKL94] Avrim Blum, Merrick L. Furst, Michael J. Kearns, and Richard J. Lipton. Cryptographic primitives based on hard learning problems. In Douglas R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 278–291. Springer, Berlin, Heidelberg, August 1994. 1, 3, 17

[BFO08] Alexandra Boldyreva, Serge Fehr, and Adam O'Neill. On notions of security for deterministic encryption, and efficient constructions without random oracles. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 335–359. Springer, Berlin, Heidelberg, August 2008. 5

[BHK11] Mark Braverman, Avinatan Hassidim, and Yael Tauman Kalai. Leaky pseudo-entropy functions. In Bernard Chazelle, editor, *Innovations in Computer Science - ICS 2011, Tsinghua University, Beijing, China, January 7-9, 2011. Proceedings*, pages 353–366. Tsinghua University Press, 2011. 5

[BHY09] Mihir Bellare, Dennis Hofheinz, and Scott Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 1–35. Springer, Berlin, Heidelberg, April 2009. 5

[BKV19] Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. CSI-FiSh: Efficient isogeny based signatures through class group computations. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part I*, volume 11921 of *LNCS*, pages 227–247. Springer, Cham, December 2019. 6

[BKW00] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. In *32nd ACM STOC*, pages 435–440. ACM Press, May 2000. 25

[BL05] Thierry P. Berger and Pierre Loidreau. How to mask the structure of codes for a cryptographic use. *Des. Codes Cryptogr.*, 35(1):63–79, 2005. 4

[BL17] Xavier Boyen and Qinyi Li. All-but-many lossy trapdoor functions from lattices and applications. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part III*, volume 10403 of *LNCS*, pages 298–331. Springer, Cham, August 2017. 23

[BLP10] Daniel J Bernstein, Tanja Lange, and Christiane Peters. Wild mceliece. In *International Workshop on Selected Areas in Cryptography*, pages 143–158. Springer, 2010. 4

[BLP11] Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Wild McEliece incognito. In Bo-Yin Yang, editor, *Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011*, pages 244–254. Springer, Berlin, Heidelberg, November / December 2011. 4

[BLPS11] Daniel J. Bernstein, Tanja Lange, Christiane Peters, and Peter Schwabe. Really fast syndrome-based hashing. In Abderrahmane Nitaj and David Pointcheval, editors, *AFRICACRYPT 11*, volume 6737 of *LNCS*, pages 134–152. Springer, Berlin, Heidelberg, July 2011. 5

[BLSV18] Zvika Brakerski, Alex Lombardi, Gil Segev, and Vinod Vaikuntanathan. Anonymous IBE, leakage resilience and circular security from new assumptions. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 535–564. Springer, Cham, April / May 2018. 2, 6, 14

[BLVW19] Zvika Brakerski, Vadim Lyubashevsky, Vinod Vaikuntanathan, and Daniel Wichs. Worst-case hardness for LPN and cryptographic hashing via code smoothing. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 619–635. Springer, Cham, May 2019. 2, 5, 7, 11, 19

[BMPS20] Jean-Francc̈ois Biasse, Giacomo Micheli, Edoardo Persichetti, and Paolo Santini. LESS is more: Code-based signatures without syndromes. In Abderrahmane Nitaj and Amr M. Youssef, editors, *AFRICACRYPT 20*, volume 12174 of *LNCS*, pages 45–65. Springer, Cham, July 2020. 2

[BQ09] Andrej Bogdanov and Youming Qiao. On the security of goldreich's one-way function. In Irit Dinur, Klaus Jansen, Joseph Naor, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 12th International Workshop, APPROX 2009, and 13th International Workshop, RANDOM 2009, Berkeley, CA, USA, August 21-23, 2009. Proceedings*, volume 5687 of *Lecture Notes in Computer Science*, pages 392–405. Springer, 2009. 3, 26

[BQ12] Andrej Bogdanov and Youming Qiao. On the security of goldreich's one-way function. *Comput. Complex.*, 21(1):83–127, 2012. 3, 26

[Bra18] Zvika Brakerski. Quantum FHE (almost) as secure as classical. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part III*, volume 10993 of *LNCS*, pages 67–95. Springer, Cham, August 2018. 1

[BV11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In Rafail Ostrovsky, editor, *52nd FOCS*, pages 97–106. IEEE Computer Society Press, October 2011. 1

[BY91] Gilles Brassard and Moti Yung. One-way group actions. In Alfred J. Menezes and Scott A. Vanstone, editors, *CRYPTO'90*, volume 537 of *LNCS*, pages 94–107. Springer, Berlin, Heidelberg, August 1991. 6

[CD23a] Wouter Castryck and Thomas Decru. An efficient key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 423–447. Springer, Cham, April 2023. 1

[CD23b] Geoffroy Couteau and Clément Ducros. Pseudorandom correlation functions from variable-density LPN, revisited. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *PKC 2023, Part II*, volume 13941 of *LNCS*, pages 221–250. Springer, Cham, May 2023. 5, 25

[CDG+17] Chongwon Cho, Nico Döttling, Sanjam Garg, Divya Gupta, Peihan Miao, and Antigoni Polychroniadou. Laconic oblivious transfer and its applications. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 33–65. Springer, Cham, August 2017. 6

[CDM+18] Geoffroy Couteau, Aurélien Dupin, Pierrick Méaux, Mélissa Rossi, and Yann Rotella. On the concrete security of Goldreich's pseudorandom generator. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part II*, volume 11273 of *LNCS*, pages 96–124. Springer, Cham, December 2018. 3, 26

[CEMT09] James Cook, Omid Etesami, Rachel Miller, and Luca Trevisan. Goldreich's one-way function candidate and myopic backtracking algorithms. In Omer Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 521–538. Springer, Berlin, Heidelberg, March 2009. 3, 26

[CGKS95] Benny Chor, Oded Goldreich, Eyal Kushilevitz, and Madhu Sudan. Private information retrieval. In *36th FOCS*, pages 41–50. IEEE Computer Society Press, October 1995. 13

[CJJ22] Arka Rai Choudhuri, Abhishek Jain, and Zhengzhong Jin. SNARGs for $\mathcal{P}$ from LWE. In *62nd FOCS*, pages 68–79. IEEE Computer Society Press, February 2022. 1

[CLM+18] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: An efficient post-quantum commutative group action. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part III*, volume 11274 of *LNCS*, pages 395–427. Springer, Cham, December 2018. 1, 6

[CLRS22] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2022. 15

[CM01] Mary Cryan and Peter Bro Miltersen. On pseudorandom generators in NC. In Jiri Sgall, Ales Pultr, and Petr Kolman, editors, *Mathematical Foundations of Computer Science 2001, 26th International Symposium, MFCS 2001 Marianske Lazne, Czech Republic, August 27-31, 2001, Proceedings*, volume 2136 of *Lecture Notes in Computer Science*, pages 272–284. Springer, 2001. 3, 26

[COT14] Alain Couvreur, Ayoub Otmani, and Jean-Pierre Tillich. Polynomial time attack on wild McEliece over quadratic extensions. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 17–39. Springer, Berlin, Heidelberg, May 2014. 4

[Cou06] Jean-Marc Couveignes. Hard homogeneous spaces. Cryptology ePrint Archive, Report 2006/291, 2006. 1, 6

[CPW20] Suvradip Chakraborty, Manoj Prabhakaran, and Daniel Wichs. Witness maps and applications. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part I*, volume 12110 of *LNCS*, pages 220–246. Springer, Cham, May 2020. 23

[CRR21] Geoffroy Couteau, Peter Rindal, and Srinivasan Raghuraman. Silver: Silent VOLE and oblivious transfer from hardness of decoding structured LDPC codes. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part III*, volume 12827 of *LNCS*, pages 502–534, Virtual Event, August 2021. Springer, Cham. 5, 25, 26

[CS02] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 45–64. Springer, Berlin, Heidelberg, April / May 2002. 6

[DGH+20] Nico Döttling, Sanjam Garg, Mohammad Hajiabadi, Daniel Masny, and Daniel Wichs. Two-round oblivious transfer from CDH or LPN. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 768–797. Springer, Cham, May 2020. 1

[DIJL23] Quang Dao, Yuval Ishai, Aayush Jain, and Huijia Lin. Multi-party homomorphic secret sharing and sublinear MPC from sparse LPN. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part II*, volume 14082 of *LNCS*, pages 315–348. Springer, Cham, August 2023. 26, 27

[DMN12] Nico Döttling, Jörn Müller-Quade, and Anderson C. A. Nascimento. IND-CCA secure cryptography based on a variant of the LPN problem. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 485–503. Springer, Berlin, Heidelberg, December 2012. 1

[DVW20] Yevgeniy Dodis, Vinod Vaikuntanathan, and Daniel Wichs. Extracting randomness from extractor-dependent sources. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 313–342. Springer, Cham, May 2020. 5

[EKM17] Andre Esser, Robert Kübler, and Alexander May. LPN decoded. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 486–514. Springer, Cham, August 2017. 25

[Fei02] Uriel Feige. Relations between average case complexity and approximation complexity. In *34th ACM STOC*, pages 534–543. ACM Press, May 2002. 3

[FGS07] Matthieu Finiasz, Philippe Gaborit, and Nicolas Sendrier. Improved fast syndrome based cryptographic hash functions. In *Proceedings of ECRYPT Hash Workshop*, volume 2007, page 155. Citeseer, 2007. 5

[FKO06] Uriel Feige, Jeong Han Kim, and Eran Ofek. Witnesses for non-satisfiability of dense random 3CNF formulas. In *47th FOCS*, pages 497–508. IEEE Computer Society Press, October 2006. 3

[FR23] Marc Fischlin and Felix Rohrbach. Searching for ELFs in the cryptographic forest. In Guy N. Rothblum and Hoeteck Wee, editors, *TCC 2023, Part III*, volume 14371 of *LNCS*, pages 207–236. Springer, Cham, November / December 2023. 3

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 169–178. ACM Press, May / June 2009. 1

[GGH11] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Collision-free hashing from lattice problems. In Oded Goldreich, editor, *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation - In Collaboration with Lidor Avigad, Mihir Bellare, Zvika Brakerski, Shafi Goldwasser, Shai Halevi, Tali Kaufman, Leonid Levin, Noam Nisan, Dana Ron, Madhu Sudan, Luca Trevisan, Salil Vadhan, Avi Wigderson, David Zuckerman*, volume 6650 of *Lecture Notes in Computer Science*, pages 30–39. Springer, 2011. 5

[Gir90] Marc Girault. A (non-practical) three-pass identification protocol using coding theory. In Jennifer Seberry and Josef Pieprzyk, editors, *AUSCRYPT'90*, volume 453 of *LNCS*, pages 265–272. Springer, Berlin, Heidelberg, January 1990. 2

[GKK20] Ankit Garg, Yael Tauman Kalai, and Dakshita Khurana. Low error efficient computational extractors in the CRS model. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 373–402. Springer, Cham, May 2020. 5

[GKM22] Venkatesan Guruswami, Pravesh K Kothari, and Peter Manohar. Algorithms and certificates for boolean csp refutation: smoothed is no harder than random. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 678–689, 2022. 26

[GKW17] Rishab Goyal, Venkata Koppula, and Brent Waters. Lockable obfuscation. In Chris Umans, editor, *58th FOCS*, pages 612–621. IEEE Computer Society Press, October 2017. 1

[GKW18] Rishab Goyal, Venkata Koppula, and Brent Waters. Collusion resistant traitor tracing from learning with errors. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *50th ACM STOC*, pages 660–670. ACM Press, June 2018. 1

[Gol00] Oded Goldreich. Candidate one-way functions based on expander graphs. *Electronic Colloquium on Computational Complexity (ECCC)*, 7(90), 2000. 3, 26

[Gol11] Oded Goldreich. Candidate one-way functions based on expander graphs. *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation: In Collaboration with Lidor Avigad, Mihir Bellare, Zvika Brakerski, Shafi Goldwasser, Shai Halevi, Tali Kaufman, Leonid Levin, Noam Nisan, Dana Ron, Madhu Sudan, Luca Trevisan, Salil Vadhan, Avi Wigderson, David Zuckerman*, pages 76–87, 2011. 1

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008. 23

[GRS12] Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. Essential coding theory. *Draft available at http://www. cse. buffalo. edu/atri/courses/coding-theory/book*, 2(1), 2012. 16

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 75–92. Springer, Berlin, Heidelberg, August 2013. 1, 23

[GVW13] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 545–554. ACM Press, June 2013. 1

[GVW15] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Predicate encryption for circuits from LWE. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 503–523. Springer, Berlin, Heidelberg, August 2015. 1

[HK12] Shai Halevi and Yael Tauman Kalai. Smooth projective hashing and two-message oblivious transfer. *Journal of Cryptology*, 25(1):158–193, January 2012. 6

[HKM23] Jun-Ting Hsieh, Pravesh K Kothari, and Sidhanth Mohanty. A simple and sharper proof of the hypergraph moore bound. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2324–2344. SIAM, 2023. 26

[HLOV11] Brett Hemenway, Benoît Libert, Rafail Ostrovsky, and Damien Vergnaud. Lossy encryption: Constructions from general assumptions and efficient selective opening chosen ciphertext security. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 70–88. Springer, Berlin, Heidelberg, December 2011. 5, 23

[Hof12] Dennis Hofheinz. All-but-many lossy trapdoor functions. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 209–227. Springer, Berlin, Heidelberg, April 2012. 23

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In *Third Algorithmic Number Theory Symposium (ANTS)*, volume 1423 of *LNCS*, pages 267–288. Springer, June 1998. 1

[HR05] Thomas Holenstein and Renato Renner. One-way secret-key agreement and applications to circuit polarization and immunization of public-key encryption. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 478–493. Springer, Berlin, Heidelberg, August 2005. 12

[HSEA14] R. Hooshmand, M. Koochak Shooshtari, T. Eghlidos, and M. R. Aref. Reducing the key length of mceliece cryptosystem using polar codes. In *2014 11th International ISC Conference on Information Security and Cryptology*, pages 104–108, 2014. 4

[IKOS08] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography with constant computational overhead. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 433–442. ACM Press, May 2008. 27

[Imp95] Russell Impagliazzo. A personal view of average-case complexity. In *Proceedings of Structure in Complexity Theory. Tenth Annual IEEE Conference*, pages 134–147. IEEE, 1995. 1

[Jab01] A Al Jabri. A statistical decoding algorithm for general linear block codes. In *Cryptography and Coding: 8th IMA International Conference Cirencester, UK, December 17–19, 2001 Proceedings 8*, pages 1–8. Springer, 2001. 25

[JLS21] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In Samir Khuller and Virginia Vassilevska Williams, editors, *53rd ACM STOC*, pages 60–73. ACM Press, June 2021. 1

[JLS22] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from LPN over $\mathbb{F}_p$, DLIN, and PRGs in NC^0. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part I*, volume 13275 of *LNCS*, pages 670–699. Springer, Cham, May / June 2022. 1

[JM96] Heeralal Janwa and Oscar Moreno. Mceliece public key cryptosystems using algebraic-geometric codes. *Designs, Codes and Cryptography*, 8(3):293–307, 1996. 4

[Jus72] Jørn Justesen. Class of constructive asymptotically good algebraic codes. *IEEE Transactions on information theory*, 18(5):652–656, 1972. 11

[KMOW17] Pravesh K. Kothari, Ryuhei Mori, Ryan O'Donnell, and David Witmer. Sum of squares lower bounds for refuting any CSP. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *49th ACM STOC*, pages 132–145. ACM Press, June 2017. 3, 26

[KMP14] Eike Kiltz, Daniel Masny, and Krzysztof Pietrzak. Simple chosen-ciphertext security from low-noise LPN. In Hugo Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 1–18. Springer, Berlin, Heidelberg, March 2014. 1, 11, 23

[KO97] Eyal Kushilevitz and Rafail Ostrovsky. Replication is NOT needed: SINGLE database, computationally-private information retrieval. In *38th FOCS*, pages 364–373. IEEE Computer Society Press, October 1997. 13, 14

[KOS10] Eike Kiltz, Adam O'Neill, and Adam Smith. Instantiability of RSA-OAEP under chosen-plaintext attack. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 295–313. Springer, Berlin, Heidelberg, August 2010. 5

[LJ12] Carl Löndahl and Thomas Johansson. A new version of mceliece pkc based on convolutional codes. In *Information and Communications Security: 14th International Conference, ICICS 2012, Hong Kong, China, October 29-31, 2012. Proceedings 14*, pages 461–470. Springer, 2012. 4

[LMPR08] Vadim Lyubashevsky, Daniele Micciancio, Chris Peikert, and Alon Rosen. SWIFFT: A modest proposal for FFT hashing. In Kaisa Nyberg, editor, *FSE 2008*, volume 5086 of *LNCS*, pages 54–72. Springer, Berlin, Heidelberg, February 2008. 5

[LMW23] Wei-Kai Lin, Ethan Mook, and Daniel Wichs. Doubly efficient private information retrieval and fully homomorphic RAM computation from ring LWE. In Barna Saha and Rocco A. Servedio, editors, *55th ACM STOC*, pages 595–608. ACM Press, June 2023. 1

[LNP22] Benoît Libert, Ky Nguyen, and Alain Passelègue. Cumulatively all-lossy-but-one trapdoor functions from standard assumptions. Cryptology ePrint Archive, Report 2022/1229, 2022. 23

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 1–23. Springer, Berlin, Heidelberg, May / June 2010. 1

[LSS23] Paul Lou, Amit Sahai, and Varun Sivashankar. Relinearization Attack On LPN Over Large Fields. *The Computer Journal*, page bxad070, 07 2023. 29

[LSSS17] Benoît Libert, Amin Sakzad, Damien Stehlé, and Ron Steinfeld. All-but-many lossy trapdoor functions and selective opening chosen-ciphertext security from LWE. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part III*, volume 10403 of *LNCS*, pages 332–364. Springer, Cham, August 2017. 23

[LT13] Grégory Landais and Jean-Pierre Tillich. An efficient attack of a McEliece cryptosystem variant based on convolutional codes. In Philippe Gaborit, editor, *Post-Quantum Cryptography - 5th International Workshop, PQCrypto 2013*, pages 102–117. Springer, Berlin, Heidelberg, June 2013. 4

[Mah18a] Urmila Mahadev. Classical homomorphic encryption for quantum circuits. In Mikkel Thorup, editor, *59th FOCS*, pages 332–338. IEEE Computer Society Press, October 2018. 1

[Mah18b] Urmila Mahadev. Classical verification of quantum computations. In Mikkel Thorup, editor, *59th FOCS*, pages 259–267. IEEE Computer Society Press, October 2018. 1

[Mal24] Giulio Malavolta. Personal communication. Email to the author, May 2024. 6

[McE78] Robert J. McEliece. A public-key cryptosystem based on algebraic coding theory. The deep space network progress report 42-44, Jet Propulsion Laboratory, California Institute of Technology, January/February 1978. `https://ipnpr.jpl.nasa.gov/progress_report2/42-44/44N.PDF`. 4, 8

[MMP+23] Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski. A direct key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 448–471. Springer, Cham, April 2023. 1

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 700–718. Springer, Berlin, Heidelberg, April 2012. 11, 16, 23

[MS07] Lorenz Minder and Amin Shokrollahi. Cryptanalysis of the sidelnikov cryptosystem. In Moni Naor, editor, *EUROCRYPT 2007*, volume 4515 of *LNCS*, pages 347–360. Springer, Berlin, Heidelberg, May 2007. 4

[MST03] Elchanan Mossel, Amir Shpilka, and Luca Trevisan. On e-biased generators in NC0. In *44th FOCS*, pages 136–145. IEEE Computer Society Press, October 2003. 3, 26

[MTSB12] Rafael Misoczki, Jean-Pierre Tillich, Nicolas Sendrier, and Paulo S. L. M. Barreto. MDPC-McEliece: New McEliece variants from moderate density parity-check codes. Cryptology ePrint Archive, Report 2012/409, 2012. 4

[MVO91] Alfred Menezes, Scott A. Vanstone, and Tatsuaki Okamoto. Reducing elliptic curve logarithms to logarithms in a finite field. In *23rd ACM STOC*, pages 80–89. ACM Press, May 1991. 1

[MW20] Tal Moran and Daniel Wichs. Incompressible encodings. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part I*, volume 12170 of *LNCS*, pages 494–523. Springer, Cham, August 2020. 5

[Nie86] H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Problems of Control and Information Theory*, 15(2):159–166, 1986. 4

[NP01] Moni Naor and Benny Pinkas. Efficient oblivious transfer protocols. In S. Rao Kosaraju, editor, *12th SODA*, pages 448–457. ACM-SIAM, January 2001. 6

[OSS84] H. Ong, Claus-Peter Schnorr, and Adi Shamir. Efficient signature schemes based on polynomial equations. In G. R. Blakley and David Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 37–46. Springer, Berlin, Heidelberg, August 1984. 1

[OTD10] Ayoub Otmani, Jean-Pierre Tillich, and Léonard Dallot. Cryptanalysis of two mceliece cryptosystems based on quasi-cyclic codes. *Mathematics in Computer Science*, 3:129–140, 2010. 4

[OW14] Ryan O'Donnell and David Witmer. Goldreich's PRG: evidence for near-optimal polynomial stretch. In *IEEE 29th Conference on Computational Complexity, CCC 2014, Vancouver, BC, Canada, June 11-13, 2014*, pages 1–12. IEEE Computer Society, 2014. 3, 26

[PR97] E. Petrank and R.M. Roth. Is code equivalence easy to decide? *IEEE Transactions on Information Theory*, 43(5):1602–1604, 1997. 2

[Pra62] Eugene Prange. The use of information sets in decoding cyclic codes. *IRE Transactions on Information Theory*, 8(5):5–9, 1962. 13, 25

[PRS12] Krzysztof Pietrzak, Alon Rosen, and Gil Segev. Lossy functions do not amplify well. In Ronald Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 458–475. Springer, Berlin, Heidelberg, March 2012. 12

[PS19] Chris Peikert and Sina Shiehian. Noninteractive zero knowledge for NP from (plain) learning with errors. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 89–114. Springer, Cham, August 2019. 1

[PVW08] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 554–571. Springer, Berlin, Heidelberg, August 2008. 6

[PW08] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 187–196. ACM Press, May 2008. 2, 4, 5, 9, 11, 19

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005. 1

[Rob23] Damien Robert. Breaking SIDH in polynomial time. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 472–503. Springer, Cham, April 2023. 1

[RRT23] Srinivasan Raghuraman, Peter Rindal, and Titouan Tanguy. Expand-convolute codes for pseudorandom correlation generators from LPN. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part IV*, volume 14084 of *LNCS*, pages 602–632. Springer, Cham, August 2023. 5, 25

[RS06] Alexander Rostovtsev and Anton Stolbunov. Public-Key Cryptosystem Based On Isogenies. Cryptology ePrint Archive, Report 2006/145, 2006. 1

[Sid94] Vladimir Michilovich Sidelnikov. A public-key cryptosystem based on binary reed-muller codes. *Discrete Mathematics and Applications*, 1994. 4

[SK14] Sujan Raj Shrestha and Young-Sik Kim. New mceliece cryptosystem based on polar codes as a candidate for post-quantum cryptography. In *2014 14th International Symposium on Communications and Information Technologies (ISCIT)*, pages 368–372, 2014. 4

[SS92] V. M. SIDELNIKOV and S. O. SHESTAKOV. On insecurity of cryptosystems based on generalized reed-solomon codes. *Discrete Mathematics and Applications*, 2(4):439–444, 1992. 4

[SV97] Amit Sahai and Salil P. Vadhan. A complete promise problem for statistical zero-knowledge. In *38th FOCS*, pages 448–457. IEEE Computer Society Press, October 1997. 2

[Wie10] Christian Wieschebrink. Cryptanalysis of the niederreiter public key scheme based on GRS subcodes. In Nicolas Sendrier, editor, *The Third International Workshop on Post-Quantum Cryptography, PQCRYPTO 2010*, pages 61–72. Springer, Berlin, Heidelberg, May 2010. 4

[WZ17] Daniel Wichs and Giorgos Zirdelis. Obfuscating compute-and-compare programs under LWE. In Chris Umans, editor, *58th FOCS*, pages 600–611. IEEE Computer Society Press, October 2017. 1

[YZ16] Yu Yu and Jiang Zhang. Cryptography with auxiliary input and trapdoor from constant-noise LPN. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 214–243. Springer, Berlin, Heidelberg, August 2016. 1

[YZW+19] Yu Yu, Jiang Zhang, Jian Weng, Chun Guo, and Xiangxue Li. Collision resistant hashing from sub-exponential learning parity with noise. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part II*, volume 11922 of *LNCS*, pages 3–24. Springer, Cham, December 2019. 2, 5, 11, 19

[Zha16] Mark Zhandry. The magic of ELFs. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 479–508. Springer, Berlin, Heidelberg, August 2016. 5

[Zha22] Mark Zhandry. New constructions of collapsing hashes. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part III*, volume 13509 of *LNCS*, pages 596–624. Springer, Cham, August 2022. 2