Collision Resistant Hashing from Learning Parity with Noise

Abstract. The Learning Parity with Noise (LPN) problem has recently found many cryptographic applications such as authentication protocols, pseudorandom generators/functions and even asymmetric tasks including public-key encryption (PKE) schemes and oblivious transfer (OT) protocols. It however remains a long-standing open problem whether LPN implies collision resistant hash (CRH) functions. Based on the recent work of Applebaum et al. (ITCS 2017), we introduce a general framework for constructing CRH from LPN for various parameter choices. We show that, just to mention a few notable ones, under any of the following hardness assumptions (for the two most common variants of LPN)

1. constant-noise LPN is $2^{n^{0.5+\varepsilon}}$-hard for any constant $\varepsilon > 0$;
2. constant-noise LPN is $2^{\Omega(n/\log n)}$-hard given $q = \mathsf{poly}(n)$ samples;
3. low-noise LPN (of noise rate $1/\sqrt{n}$) is $2^{\Omega(\sqrt{n}/\log n)}$-hard given $q = \mathsf{poly}(n)$ samples.

there exists CRH functions with constant (or even poly-logarithmic) shrinkage, which can be implemented using polynomial-size depth-3 circuits with NOT, (unbounded fan-in) AND and XOR gates. Our technical route LPN→bSVP→CRH is reminiscent of the known reductions for the large-modulus analogue, i.e., LWE→SIS→CRH, where the binary Shortest Vector Problem (bSVP) was recently introduced by Applebaum et al. (ITCS 2017) that enables CRH in a similar manner to Ajtai's CRH functions based on the Short Integer Solution (SIS) problem.

Furthermore, under certain additional (arguably minimal) idealized assumptions, such as small-domain random functions or that a block cipher (keyed by a public random string) behaves like a random permutation, we obtain more efficient and polynomially shrinking CRH functions from $2^{n^{0.5+\varepsilon}}$-hard constant-noise LPN or $2^{n^{0.25+\varepsilon}}$-hard low-noise LPN. In particular, the construction of hash functions follows a conceptually simple approach: it divides its input into many equal-length blocks, evaluates random functions (or blockciphers) on them independently and in parallel, and then produces their XOR sum as output.

Keywords: Foundations, Learning Parity with Noise, Binary Shortest Vector Problem, Collision Resistant Hash Functions.

1 Introduction

1.1 Learning Parity with Noise

Learning Parity with Noise. The computational version of the Learning Parity with Noise (LPN) assumption with secret size $n \in \mathbb{N}$ and noise rate $0 < \mu < 1/2$ postulates that given any number of samples $q = \mathsf{poly}(n)$ it is computationally infeasible for any probabilistic polynomial-time (PPT) algorithm to recover the random secret $\mathbf{x} \xleftarrow{\$} \{0,1\}^n$ given $(\mathbf{A},\ \mathbf{A} \cdot \mathbf{x} + \mathbf{e})$, where $\mathbf{A}$ is a random $q \times n$ Boolean matrix, $\mathbf{e}$ follows $\mathcal{B}_\mu^q = (\mathcal{B}_\mu)^q$, $\mathcal{B}_\mu$ denotes the Bernoulli distribution with parameter μ (taking the value 1 with probability μ and the value 0 with probability $1-\mu$), '$\cdot$' and '$+$' denote (matrix-vector) multiplication and addition over GF(2) respectively. The decisional version of LPN simply assumes that $(\mathbf{A},\ \mathbf{A} \cdot \mathbf{x} + \mathbf{e})$ is pseudorandom. The two versions are polynomially equivalent [14,40,6].

Hardness of LPN. The computational LPN problem can be seen as the average-case analogue of the NP-complete problem "decoding random linear codes" [10]. LPN has been also extensively studied in learning theory, and it was shown in [34] that an efficient algorithm for LPN would allow to learn several important function classes such as 2-DNF formulas, juntas, and any function with a sparse Fourier spectrum. When the noise rate μ is constant (i.e., independent of secret size n), Blum, Kalai and Wasserman [15] showed how to solve LPN with time/sample complexity $2^{O(n/\log n)}$. Lyubashevsky [45] observed that one can produce almost as many LPN samples as needed using only $q = n^{1+\epsilon}$ LPN samples (of a lower noise rate), which implies a variant of the BKW attack [15] with time complexity $2^{O(n/\log\log n)}$ and sample complexity $n^{1+\epsilon}$. If one is restricted to $q = O(n)$ samples, then the best attack has exponential complexity $2^{O(n)}$ [50]. Under low noise rate $\mu = 1/\sqrt{n}$, the best attacks [19,11,43,9] solve LPN with time complexity $2^{O(\sqrt{n})}$. The low-noise LPN is mostly believed a stronger assumption than constant-noise LPN. In noise regime $\mu = 1/\sqrt{n}$, LPN can be used to build public-key encryption (PKE) schemes [2] and oblivious transfer (OT) protocols. Quantum algorithms [33] that build upon Grover search may achieve a certain level (up to quadratic) of speedup over classic ones in solving LPN, which does not change the asymptotic order (up to the constant in the exponent) of the complexity of the problem. This makes LPN a promising candidate for "post-quantum cryptography". Furthermore, LPN enjoys simplicity and is more suited for weak-power devices (e.g., RFID tags) than other quantum-secure candidates such as Learning with Errors (LWE) [57] as the many modular additions and multiplications in LWE would be simplified to AND and XOR gates in LPN.

Symmetric-key cryptography from constant-noise LPN. LPN was used to build lightweight authentication schemes (e.g. [36,39,40], just to name a few). Kiltz et al. [42] and Dodis et al. [27] constructed randomized MACs from LPN, which implies a two-round authentication scheme with security against

active adversaries. Lyubashevsky and Masny [47] gave a more efficient three-round authentication scheme from LPN and recently Cash, Kiltz, and Tessaro [20] reduced the round complexity to 2 rounds. Applebaum et al. [4] used LPN to construct efficient symmetric encryption schemes with certain key-dependent message (KDM) security. Jain et al. [38] constructed an efficient perfectly binding string commitment scheme from LPN. We refer to the survey [55] about cryptography from LPN.

Public-key cryptography and more from low-noise LPN. Alekhnovich [2] established the feasibility result that public-key encryption (PKE) can be based on LPN in the low-noise regime of $\mu = 1/\sqrt{n}$. Döttling et al. [31] and Kiltz et al. [41] further showed that low-noise LPN alone already suffices for PKE schemes with CCA (and KDM [30]) security. Once we obtain a PKE, it is perhaps not so surprising to build an oblivious transfer (OT) protocol. That is, LPN-based PKE uses pseudorandom public keys (so that one can efficiently fake random public keys that are computationally indistinguishable from real ones) and in this scenario Gertner et al. [35] showed how to construct an OT protocol in a black-box manner. This observation was made explicit in [24], where universally composable OT protocols were constructed from low-noise LPN. All the above schemes are based on LPN of noise rate $1/\sqrt{n}$. The only exception seems to be the recent result by Yu and Zhang [63] that PKE and OT can also be based on constant-noise LPN with hardness $2^{n^{1/2+\epsilon}}$.

Open problems and recent progress. It remains open [55,46] whether LPN implies other advanced cryptographic objects, such as fully homomorphic encryption (FHE) and collision resistant hash (CRH) functions. Brakerski [16] reported some negative result that straightforward LPN-based encryptions are unlikely to achieve full homomorphism. As for LPN-based CRH, a notable progress was recently made by Applebaum et al. [5], who showed that $2^{\Omega(n/\log n)}$-hard constant-noise LPN implies CRH [1]. Based on some ideas (in particular, the bSVP assumption) from [5], we introduce a general framework of constructing CRH from LPN with various tunable trade-offs between the parameters (e.g., noise rate, hardness, shrinkage), and then present the main feasibility results in commonly assumed noise regimes.

On the concurrent work of [18]. To our best knowledge[2], Brakerski et al. [18] constructed CRH from LPN at the (extremely low) noise rate of $\mu = \log^2 n/n$, which can be derived as a special case under our framework.

[1] More precisely, [5] obtains a win-win result that either constant-noise LPN implies CRH or one achieves arbitrary polynomial speedup over the BKW algorithm [15].

[2] [18] hasn't been publicly available yet (by the end of 2017) and the only information we learn about [18] is through [17], where Sections 1.1 and 2.3 of [17] sketch the CRH construction of [18] assuming LPN with an extremely low noise rate.

1.2 Cryptographic Hash Functions

A CRYPTOGRAPHIC HASH FUNCTION $\{0,1\}^* \to \{0,1\}^n$ is a deterministic function that maps arbitrarily (or at least sufficiently) long bit strings into digests of a fixed length. The function was originally introduced in the seminal work of Diffie and Hellman [26] to produce more efficient and compact digital signatures. As exemplified by MD5 and SHA-1/2/3, it is now one of the most widely used cryptographic primitives in security applications and protocols, such as SSL/TLS, PGP, SSH, S/MIME, IPsec and Bitcoin. Merkle [53] formulated three main security properties (that still remain in use to date) of a cryptographic hash function: preimage resistance, second preimage resistance and collision resistance, of which collision resistance seems the most essential and suffices for many aforementioned applications [3]. Similar to the mode of operations for data encryption, the design of cryptographic hash functions proceeds in two steps: one first designs a compression function that operates on fixed-length inputs and outputs, and then applies a domain extender to accept messages of arbitrary length. This dates back to the independent work of Merkle [54] and Damgård [23], who proposed a domain extender, and showed that if the underlying compression function is collision resistant then so is the hash function based on the Merkle-Damgård construction. We refer to [3] for a survey about various domain extenders for cryptographic hash functions. For the rest of this paper we will focus on such length-regular compression functions, namely, CRH functions.

COLLISION RESISTANT HASHING. Theoretical constructions of CRH functions can be based on the hardness of factoring and discrete logarithm (via the construction of claw-free permutations [22]), which are however far from practical. Ajtai [1] introduced an elegant and (conceptually) simple construction: $f_{\mathbf{A}} : \{0,1\}^m \to \mathbb{Z}_p^n$ that for a random $\mathbf{A} \in \mathbb{Z}_p^{n \times m}$ and some (at least polynomially) large p and on input $\mathbf{z} \in \{0,1\}^m$ it computes

$$f_{\mathbf{A}}(\mathbf{z}) = \mathbf{A} \cdot \mathbf{z} \mod p \ , \tag{1}$$

which is collision resistant via a security reduction from the Short Integer Solution (SIS) problem, and is thus at least as hard as lattice problems such as GapSVP and SIVP. Lyubashevsky et al. [48] gave a ring-based variant of Ajtai's construction, called SWIFFT, which admits FFT and precomputation techniques for improved efficiency while preserves an asymptotic security proof from ideal lattices at the same time. Despite a substantial gap between the claimed security level and the actual security bounds proved, SWIFFT [48] and its modified version (as a SHA-3 candidate) SWIFFTX [7] are among the very few hash function designs combining the best of two worlds (i.e., practical efficiency and rigorous security proof).

[3] Unlikely collision resistance whose definition is unique and unambiguous, there are several variants of (second) preimage resistance for which people strive to find a compromise that facilitates security proofs yet captures the needs of most applications. Some variants of (second) preimage resistance are implied by collision resistance in the conventional or provisional sense [59].

The Expand-then-Compress Approach. Recently, Applebaum et al. [5] constructed a function $h_{\mathbf{M}} : \{0,1\}^k \to \{0,1\}^n$ keyed by a random $n \times q$ binary matrix $\mathbf{M}$ as:

$$h_{\mathbf{M}}(\mathbf{y}) = \mathbf{M} \cdot \mathsf{Expand}(\mathbf{y}) \ , \tag{2}$$

where Expand is an injective function that expands $\mathbf{y}$ into a much longer yet sparse string, i.e., for every $\mathbf{y} \in \{0,1\}^k$: $t = |\mathsf{Expand}(\mathbf{y})| < n < k < q$. Note that $h_{\mathbf{M}}$ can be viewed as a binary version of Ajtai's CRH (see $f_{\mathbf{A}}$ in (1)), where matrix $\mathbf{A}$ over $\mathbb{Z}_p$ is simplified to a binary matrix $\mathbf{M}$, and binary vector $\mathbf{z}$ is further flattened to a sparse binary vector $\mathsf{Expand}(\mathbf{y})$. Thanks to the simplification to the binary field, $h_{\mathbf{M}}$ can be implemented rather efficiently both in the asymptotic sense and in practice. Under certain realizations of Expand (see Lemma 3.1), $h_{\mathbf{M}}$ (for any specified $\mathbf{M}$) can be directly translated to a polynomial-size circuit of NOT, (unbounded fan-in) XOR and AND gates in depth 3 (or even depth 2 if the input includes not only the individual bits of $\mathbf{y}$ but also their respective complements). Interestingly, the FSB hash proposal [8] and its variant the RFSB hash [12] fall into concrete (but over optimistic) instantiations of $h_{\mathbf{M}}$.[4]

Binary SVP. In order to study the asymptotic security of the EtC hash, Applebaum et al. [5] recently introduced the binary Shortest Vector Problem (binary SVP or bSVP in short). Informally, the bSVP assumption asserts that given a random matrix distribution[5] $\mathbf{M} \xleftarrow{\$} \{0,1\}^{n \times q}$, it is computationally infeasible to find a non-zero $\mathbf{x} \in \{0,1\}^q$ of Hamming weight $t \ll q$ such that $\mathbf{Mx} = \mathbf{0}$. From a code-theoretic perspective, $\mathbf{M}$ specifies the $n \times q$ parity check matrix of a random binary linear code of rate $1 - n/q$, where the rows of $\mathbf{M}$ are linearly independent (except with negligible probability), and therefore the bSVP postulates that finding a short codeword is hard in the average case. We refer to [5] for discussions about meaningful regimes of (t/q) that give rise to one-way functions and collision resistant hash functions. Similar to SIS, bSVP immediately implies CRH as any efficient algorithm that comes up with a collision $h_{\mathbf{M}}(\mathbf{y}) = h_{\mathbf{M}}(\mathbf{y}')$ for $\mathbf{y} \neq \mathbf{y}'$ immediately implies a solution to bSVP, i.e., $\mathbf{M} \cdot \mathbf{x} = \mathbf{0}$, where $\mathbf{x} = \mathsf{Expand}(\mathbf{y}) - \mathsf{Expand}(\mathbf{y}')$ has weight no greater than $2t$. We mention that in the worst case, it is NP-hard to compute (or even to approximate by a constant factor) the distance of linear code [62,32]. However, as an average-case hardness assumption, bSVP is relatively new and deserves further investigation. A shortcut and promising direction is to see whether bSVP is reducible from the learning parity with noise (LPN) problem since they are both related to random binary linear codes, and the average-case hardness of the latter is well understood. However, the work of [5] only established a weak connection between bSVP and LPN. That is, they show that at least one of the following is true:

[4] However, our results do not immediately constitute security proofs for the FSB-style hash functions as there remains a substantial gap between the security proved and security level claimed by the FSB instantiation.

[5] $\mathbf{M}$ in our consideration has dimension $n \times q$ instead of $\alpha n \times n$ considered by [5].

1. One can achieve an arbitrary polynomial speedup over the BKW algorithm [15], i.e., for every constant $\varepsilon > 0$ there exists an algorithm that solves constant-noise LPN with time and sample complexity $2^{\frac{\varepsilon n}{\log n}}$ for infinitely many n's.
2. There exist CRH functions of constant shrinkage and logarithmic degree.

Otherwise stated, assume that the BKW algorithm cannot be further improved asymptotically, then bSVP (for certain parameters) and CRH are implied.

1.3 The Framework of Building CRH from LPN

Duality between LPN and bSVP. We explain the high-level intuition of how LPN relates (and reduces) to bSVP (deferring the choices of non-trivial parameters to next paragraph), which in turn implies CRH. Under the theme of "decoding random linear codes" where row vector $\mathbf{s}^\mathsf{T}$ is the message, $\mathbf{M}$ is an $n \times q$ generator matrix and $\mathbf{s}^\mathsf{T}\mathbf{M}$ is the codeword, the idea is to use a (sparse) column vector $\mathbf{x}$ from the corresponding parity matrix such that any (noisy) codeword multiplied by $\mathbf{x}$ is (biased to) 0, regardless of the value of $\mathbf{s}$. Informally, assume for contradiction that a useful bSVP solver succeeds in finding a sparse vector $\mathbf{x}$ for an $n \times q$ matrix $\mathbf{M}$ such that $\mathbf{M}\mathbf{x} = \mathbf{0}$, then this leads to a distinguishing attack against the LPN instance $(\mathbf{M}, \mathbf{s}^\mathsf{T}\mathbf{M} + \mathbf{e}^\mathsf{T})$ by computing

$$(\mathbf{s}^\mathsf{T}\mathbf{M} + \mathbf{e}^\mathsf{T}) \cdot \mathbf{x} = \mathbf{e}^\mathsf{T}\mathbf{x}$$

which is a biased bit (and thus distinguishable from uniform) due to the sparseness of $\mathbf{x}$ and $\mathbf{e}$. This already constitutes a contradiction to the decisional LPN, and one can repeat the above on sufficiently many independent samples (with a majority voting) to gain a constant advantage, and further transform it into a key-recovery attack using the same number of samples [6].

Main feasibility results. By exploiting the duality between LPN and bSVP, we present a general framework stated in Theorem 1.1 below (and more formally in Theorem 3.1) that enables to construct CRH from LPN for various tunable parameter choices, as stated in Corollary 1.1 (and more formally in Corollary 3.1). The constructions follow the Expand-then-Compress approach and can be implemented by a polynomial-size depth-3 circuit with NOT, (unbounded fan-in) AND and XOR gates[6]. The framework, in particular, when tuned to params #2 and #4 of Corollary 1.1, encompasses the known results obtained in [6] and the concurrent work of [18]. In addition, it establishes feasibility results for constant-noise LPN assuming much less hardness (see param #1 of Corollary 1.1) and for low-noise LPN (see param #3 of Corollary 1.1), which was not previously known. We remark that the $2^{\Omega(\sqrt{n}/\log n)}$-hardness assumption for low-noise LPN is quite reasonable as the current best attacks need complexity $\mathsf{poly}(n) \cdot e^{\sqrt{n}}$ [44] for which even improving upon the constant in the exponent seems highly non-trivial. Further, the $2^{n^{0.501}}$-hardness assumed for constant-noise LPN offers even more generous security margins as the best attack goes even beyond $2^{n^{0.999}}$ [15].

[6] The circuit falls into the class $\mathsf{AC}^0(\mathsf{MOD2})$. See Section 2 for a formal definition.

Theorem 1.1 (main framework, informal). *Let n be the security parameter, and let $\mu = \mu(n)$, $k = k(n)$, $q = q(n)$, $t = t(n)$ and $T = T(n)$ such that $t \ll q$ (e.g., $t \leq \sqrt{q}$), $q \leq T \approx 2^{\frac{8\mu t}{\ln 2(1-2\mu)}}$. Assume that the (decisional) LPN problem of size n and noise rate μ is T-hard given q samples, and let*

$$h_{\mathbf{M}} : \{0,1\}^k \to \{0,1\}^n, \;\; h_{\mathbf{M}}(\mathbf{y}) = \mathbf{M} \cdot \mathsf{Expand}(\mathbf{y}), \;\; \mathsf{Expand} : \{0,1\}^k \to \{0,1\}^q,$$

be functions satisfying the following conditions:

1. *($h_{\mathbf{M}}$ **is compressing**). $k > n$;*
2. *(Expand **has sparse outputs**). for all $\mathbf{y} \in \{0,1\}^k$: $|\mathsf{Expand}(\mathbf{y})| = t$;*
3. *(Expand **is injective**). Expand is an injection with $k \approx \log \binom{q}{t} = (1 + o(1)) \log(q/t)t > t \log q/2$ (see Fact 2 and the inequality is due to $t \leq \sqrt{q}$).*

Then, $h_{\mathbf{M}}$[7] is a CRH function with shrinkage factor $\frac{n}{k}$.

Rationale. Upon any collision $\mathbf{y} \neq \mathbf{y}'$ we get that $\mathbf{x} = \mathsf{Expand}(\mathbf{y}) - \mathsf{Expand}(\mathbf{y}')$ such that $\mathbf{e}^{\mathsf{T}}\mathbf{x}$, i.e., the XOR sum of up to $2t$ bits drawn from $\mathcal{B}_\mu$ is

$$\frac{1}{2} + \frac{2^{-(\log \frac{1}{1-2\mu})2t}}{2} \geq \frac{1}{2} + \frac{2^{-\frac{4\mu t}{\ln 2(1-2\mu)}}}{2}$$

biased to 0 by the Piling up lemma (Lemma 2.1) and inequality $\ln(1+x) \leq x$. Otherwise said, the underlying decisional LPN must be $2^{\frac{\Omega(\mu t)}{(1-2\mu)}}$-hard to counteract the aforementioned attack. We refer to Theorem 3.1 for a more formal statement and a rigorous proof. The framework allows for various trade-offs between μ, q and T (via the intermediate parameter t) and we state a few notable ones in Corollary 1.1 below. Moreover, the CRH can be contained in $\mathsf{AC}^0(\mathsf{MOD2})$ based on a parallel implementation of the underlying function Expand in AC^0.

Corollary 1.1 (LPN $\to$ CRH). *Type LPN with Hardness implies CRH with Shrinkage in $\mathsf{AC}^0(\mathsf{MOD2})$, where (Type , Hardness, Shrinkage) can be (but are not limited to) any of the following:*

1. (Constant-noise, less hardness, poly-logarithmic shrinkage).
   $\mu = O(1)$, $T = 2^{n^{0.5+\varepsilon}}$, $q = 2^{n^{0.5}}$ and $\frac{n}{k} < \frac{16\mu}{\ln 2(1-2\mu)n^{\varepsilon}} = \frac{16\mu}{\ln 2(1-2\mu)\log^{2\varepsilon}\lambda}$ for any constant $\varepsilon > 0$.
2. (Constant-noise, more hardness, constant shrinkage).
   $\mu = O(1)$, $T = 2^{\frac{\varepsilon n}{\log n}}$, $q = n^{C_{\varepsilon,\mu}}$, $\frac{n}{k} < \frac{1}{2}$ for any constant $\varepsilon > 0$ and $C_{\varepsilon,\mu} = \max(\frac{32\mu}{\varepsilon \ln 2(1-\mu)}, 2)$.
3. (Low-noise, more hardness, constant shrinkage).
   $\mu = 1/\sqrt{n}$, $T = 2^{\frac{\varepsilon\sqrt{n}}{\log n}}$, $q = n^{C_{\varepsilon,\mu}}$, $\frac{n}{k} < \frac{1}{2}$ for any constant $\varepsilon > 0$ and $C'_{\varepsilon,\mu} = \max(\frac{32}{\varepsilon \ln 2}, 2)$.

[7] More strictly speaking, the resulting CRH is either $h_{\mathbf{M}}$ itself or its domain-extended version (by a parallel repetition).

4. (Extremely low-noise, standard hardness, constant shrinkage). $\mu = \frac{(\log n)^2}{n}$, $T = q > \mathsf{poly}(n)$ for every poly, and $\frac{n}{k} < \frac{1}{2}$.

Intuitions about parameters choices. The aforementioned parameter choices are not exhaustive but they follow quite naturally from the respective noise rates. We explain the underlying intuitions for making such choices (and refer to Corollary 3.1 and its proof for formal details). For immediate efficiency we set $q = \mathsf{poly}(n)$ (s.t. the dimensions of $\mathbf{M}$ are polynomially related) and constant shrinkage factor $\frac{n}{k} < \frac{2n}{t \log q} = \frac{1}{2}$, and therefore $t = \Omega(n/\log n)$ and it requires hardness $T = 2^{\Omega(\mu n/\log n)}$. This yields the parameter settings #2, #3 and #4 for constant, low and extremely low noise rates respectively. Alternatively, in favor of minimized hardness assumed for constant-noise LPN, we let the sample complexity be nearly the same as time complexity up to a factor n^{ε},[8] i.e., $\log(q) = \Omega(\frac{\log T}{n^{\varepsilon}}) = \Omega(\frac{t}{n^{\varepsilon}})$ and thus the injective condition becomes $k = \Omega(t^2/n^{\varepsilon})$ and $\frac{n}{k} < \frac{n^{1+\varepsilon}}{\Omega(t^2)}$, which results in param #1 by setting $t = \Omega(n^{0.5+\varepsilon})$. However, now the issue is that the dimensions q and n of $\mathbf{M}$ are not polynomially related and thus it does not immediately give rise to an efficient CRH. This motivates us to switch to another parameter $\lambda = q = 2^{\sqrt{n}}$ such that $h_{\mathbf{M}} : \{0,1\}^{\Omega(\log^{2+2\varepsilon} \lambda)} \to \{0,1\}^{\log^2 \lambda}$ for $\mathbf{M} \in \{0,1\}^{\log^2 \lambda \times \lambda}$ is a $\lambda^{\Omega(\log^{2\varepsilon} \lambda)}$-hard CRH function computable in time $\mathsf{poly}(\lambda)$, which further implies a domain-extended CRH $h'_{\mathbf{M}} : \{0,1\}^{\Omega(\lambda \log^{2\varepsilon} \lambda)} \to \{0,1\}^{\lambda}$ by a parallel repetition.

Feasibilities vs. limits. Admittedly, the limits of the framework are obvious: unless under extremely low noise rate [18] the hardness assumed is much beyond polynomial (although still reasonable given the current state-of-the-art). Moreover, the parameter-switching technique (that helps to reduce hardness assumed) dramatically downgrades the security and deteriorates the shrinkage factor from polynomial to poly-logarithmic. Further, the technique only applies to constant noise: if the noise rate μ depends on n, e.g., $\mu = 1/\sqrt{n}$, then switching to a new parameter, say $\lambda = 2^{n^{0.25}}$, yields lifted noise rate $\mu = 1/\log^2 \lambda$. We offer an alternative to avoid the efficiency/security loss by assuming a minimal amount of heuristics, i.e., a small domain random function. This helps to obtain more secure, polynomially shrinking and parallelizable CRH functions from the least hardness possible, e.g., it requires only $2^{n^{0.25+\varepsilon}}$-hardness for low-noise LPN.

Corollary 1.2 (LPN → CRH with idealized heuristics, informal). *Assume that (n,μ,q)-DLPN is T-hard and $R : \{0,1\}^{\log(q)} \to \{0,1\}^n$ with $\log(q) \ll n$ behaves like a random function, then for $\mathbf{y} = \mathbf{y}_1 \| \cdots \| \mathbf{y}_t$ parsed as $t = k/L$ blocks, each of size $L = \log(q/t)$, we have*

$$h^R : \{0,1\}^k \to \{0,1\}^n, \quad h^R(\mathbf{y}) = \bigoplus_{i=1}^{t} R(i \| \mathbf{y}_i),$$

[8] By switching to a new security parameter, we eventually obtain a CRH function with polynomial running time and super-polynomial security for which the n^{ε} gap factor plays a vital role.

is a CRH function with shrinkage $\frac{n}{k}$, where (μ,T,q,$\frac{n}{k}$) can be either below:

1. *(Constant-noise, less hardness, polynomial shrinkage).* $\mu = O(1)$, $T = 2^{n^{0.5+\varepsilon}}$, $q = 2^{n^{0.5}}$ *and* $\frac{n}{k} < \frac{16\mu}{\ln 2(1-2\mu)n^{\varepsilon}}$ *for any constant* $\varepsilon > 0$.
2. *(Low-noise, less hardness, polynomial shrinkage).* $\mu = 1/\sqrt{n}$, $T = 2^{n^{0.25+\varepsilon}}$, $q = 2^{n^{0.25}}$ *and* $\frac{n}{k} < \frac{16}{\ln 2 \cdot n^{\varepsilon}}$ *for any constant* $\varepsilon > 0$.

On related heuristic-based approaches. It may seem trivial to obtain CRHs from idealized heuristics such as random oracles and ideal ciphers, but we stress that we only make a quite light use of idealism by assuming a small-domain random function with inputs much shorter than outputs (for which domain extension is non-trivial), which can be efficiently instantiated from practical objects such as blockciphers (assuming that a blockcipher on a public random key behaves like a random permutation). In contrast, most previous blockcipher-based compression functions (e.g. [54,56,13]) reside in the (much stronger) Ideal Cipher Model that a block cipher on every key behaves like an independent random permutation. Moreover, existing permutation-based solutions either only offer a constant shrinkage factor (typically 1/2) [61,51], or require permutations with a large domain (e.g., [29] needs a large permutation over $\{0,1\}^{n^2}$ to obtain a CRH function with shrinkage factor $1/n$).

2 Preliminaries

Notations and definitions. Column vectors are represented by bold lowercase letters (e.g., $\mathbf{s}$), row vectors are denoted as their transpose (e.g., $\mathbf{s}^{\mathsf{T}}$), and matrices are denoted by bold capital letters (e.g., $\mathbf{A}$). $|s|$ refers to the Hamming weight of binary string s. We use $\mathcal{B}_\mu$ to denote the Bernoulli distribution with parameter μ, while $\mathcal{B}_\mu^q$ denotes the concatenation of q independent copies of $\mathcal{B}_\mu$. We use $\log(\cdot)$ to denote the binary logarithm. $\mathbf{x} \xleftarrow{\$} \mathcal{X}$ refers to drawing $\mathbf{x}$ from set $\mathcal{X}$ uniformly at random, and $\mathbf{x} \leftarrow X$ means drawing $\mathbf{x}$ according to distribution X. $\mathbf{a}\|\mathbf{b}$ denotes the concatenation of $\mathbf{a}$ and $\mathbf{b}$. A function $\mathsf{negl}(\cdot)$ is negligible if for any constant N_c we have that $\mathsf{negl}(n) < 1/\mathsf{poly}(n)$ for every polynomial poly and all sufficiently large n. AC^0 refers to the class of polynomial-size, constant-depth circuit families with unbounded fan-in AND and OR gates, where NOT gates are allowed only at input level. $\mathsf{AC}^0(\mathsf{MOD2})$ refers to the class of polynomial-size, constant-depth circuit families with unbounded fan-in AND, OR and XOR gates.

We define decisional and computational LPN problems, and we just use the decisional one due to their polynomial equivalence. In particular, there are computational-to-decisional reductions even for the same sample complexity [6].

Definition 2.1 (Learning Parity with Noise). *Let n be the security parameter, and let $\mu = \mu(n)$, $q = q(n)$ and $T = T(n)$. The decisional LPN problem with secret length n, noise rate $0 < \mu < 1/2$ and sample complexity q, denoted*

by (n,μ,q)-DLPN, is T-hard if every probabilistic algorithm $\mathcal{D}$ of running time T we have that the following holds for all sufficiently large n's

$$\left| \Pr[\mathcal{D}(\mathbf{A},\ \mathbf{A}\cdot\mathbf{x}+\mathbf{e}) = 1] - \Pr[\mathcal{D}(\mathbf{A},\mathbf{y}) = 1] \right| \leq \frac{1}{T}\ , \tag{3}$$

and the computational LPN problem with the same n, μ and q, denoted by (n,μ,q)-LPN, is T-hard if for every probabilistic algorithm $\mathcal{D}$ of running time T we have that the following holds for all sufficiently large n's

$$\Pr[\ \mathcal{D}(\mathbf{A},\ \mathbf{A}\cdot\mathbf{x}+\mathbf{e}) = \mathbf{x}\ ] \ \leq\ \frac{1}{T}\ , \tag{4}$$

where $q \times n$ matrix $\mathbf{A} \xleftarrow{\$} \{0,1\}^{q\times n}$ and $\mathbf{x} \xleftarrow{\$} \{0,1\}^n$, $\mathbf{y} \xleftarrow{\$} \{0,1\}^q$ and $\mathbf{e} \leftarrow \mathcal{B}_\mu^q$.
STANDARD HARDNESS. *We recall that standard polynomial hardness requires that $T > \mathsf{poly}(n), q > \mathsf{poly}(n)$ and for every* poly *and all sufficiently large n's.*

Unlike other primitives (such as one-way functions, pseudorandom generators and functions) whose security parameter is typically the input/key length, the security strength of collision resistant hash functions are more often represented as a function of the output length n and it is upper bounded by $2^{n/2}$ due to birthday attacks. In practice, a fixed output size (e.g. 128, 160) typically corresponds to a single function (e.g., MD5, SHA1) instead of a collection of ones [9]. One can just stick to a $h_{\mathbf{M}}$ for some pre-fixed random $\mathbf{M}$.

Definition 2.2 (Collision Resistant Hash Functions). *A collection of functions*

$$\mathcal{H} = \left\{ h_z : \{0,1\}^{k(n)} \to \{0,1\}^n, z \in \{0,1\}^{s(n)} \right\}$$

is a collision-resistant hash (CRH) function if the following hold:

- ***(Shrinking).*** *The shrinkage factor of $\mathcal{H}$, defined as ratio $\frac{n}{k}$, is less than 1 for every n.*
- ***(Efficient).*** *There are efficient algorithms H and $\mathcal{G}$: (1) on input $z \in \{0,1\}^s$ and $y \in \{0,1\}^k$, H outputs $h_z(y)$; and (2) given 1^n as input $\mathcal{G}$ returns an index $z \in \{0,1\}^s$.*
- ***(Collision-resistant).*** *For every probabilistic polynomial-time (PPT) adversary $\mathcal{A}$*

$$\Pr_{z\leftarrow\mathcal{G}(1^n)}[\ (y,y') \leftarrow \mathcal{A}(z) : y \neq y' \wedge h_z(y) = h_z(y')\ ] = \mathsf{negl}(n)\ .$$

The shrinkage is linear if $n/k \leq 1-\epsilon$, and it is poly-logarithmic (resp., polynomial) if $n/k \leq 1/\log^\epsilon n$ (resp., $n/k \leq 1/n^\epsilon$) for some positive constant $\epsilon > 0$.
T-HARDNESS. *For $T = T(n)$ we call $\mathcal{H}$ a T-hard CRH if no probabilistic adversary $\mathcal{A}$ of running time T finds any collision with probability more than $1/T$.*

[9] Recall that a non-uniform attacker can obtain polynomial-size non-uniform advice. Thus, if every security parameter corresponds to only a single function h then the attacker can include a pair of x and x' with $h(x) = h(x')$ as part of the advice.

The indifferentiability framework [49,21] is widely adopted to analyze and prove the security of the construction of one idealized primitive from another, typically in settings where the underlying building blocks have no secrets.

Definition 2.3 (Indifferentiability [21]). *A Turing machine C with oracle access to an ideal primitive P is (q, σ, t, ϵ)-indifferentiable from an ideal primitive R, if there exists a simulator S with oracle access to R such that for any distinguisher D that makes at most q queries, it holds that*

$$\left| \Pr[D^{C^P, P} = 1] - \Pr[D^{R, S^R} = 1] \right| \leq \epsilon,$$

where S makes σ queries and runs in time t when interacting with D and R.

The implication is that C^P can securely replace R in many scenarios. We refer to [58,25] for discussions on the (in)applicability of indifferentiability results.

Lemma 2.1 (Piling-up lemma). *For $0 < \mu < 1/2$ and random variables E_1, E_2, $\cdots$, E_ℓ that are i.i.d. to $\mathcal{B}_\mu$ we have*

$$\Pr\left[\bigoplus_{i=1}^{\ell} E_i = 0\right] = \frac{1}{2}(1 + (1-2\mu)^\ell) = \frac{1}{2} + 2^{-c_\mu \ell - 1} \ ,$$

where $c_\mu = \log \frac{1}{1-2\mu}$.

Fact 1 *For any $0 \leq x \leq 1$ it holds that $\log(1+x) \geq x$; and for any $x > -1$ we have $\log(1+x) \leq x / \ln 2$.*

Fact 2 *For $k = o(n)$ we have $\log \binom{n}{k} = (1 + o(1)) k \log \frac{n}{k}$.*

3 Collision Resistant Hash Functions

3.1 The Expand-then-Compress Construction

We give a high-level overview about the EtC construction from [5]. Fix a random $n \times q$ matrix $\mathbf{M}$ which specifies the function. On input $\mathbf{y}$, $h_{\mathbf{M}}$ first stretches it into a long-but-sparse vector, i.e., $\mathsf{Expand}(\mathbf{y})$, and then multiply it with $\mathbf{M}$, which compresses into n bits. There are many ways to instantiate $h_{\mathbf{M}}$ and we use the following one which fulfills all properties needed by our framework (cf. Theorem 1.1). In addition, Expand is highly parallel and can be efficiently implemented by a single layer of (unbounded fan-in) AND gates (assuming input includes both the individual bits of $\mathbf{y}$ and also their respective complements), and therefore $h_{\mathbf{M}}$ simply builds upon Expand by adding a layer of XOR gates. Furthermore, the Expand function can be efficiently instantiated with idealized heuristics (see Lemma 3.3).

Lemma 3.1 (A realization of the expanding function [5]). *Let n be the security parameter and let $k \leq \mathsf{poly}(n)$, $L = O(\log n)$, $t = t(n)$, $q = q(n)$ be integer-valued functions such that $k = L \cdot t$, $q = t \cdot 2^L$. Let $\mathsf{Expand} : \{0,1\}^k \to \{0,1\}^q$ be a function that parses the k-bit input into L-bit blocks as*

$$\mathbf{y} = y_1 \cdots y_L \| y_{L+1} \cdots y_{2L} \| \cdots \| y_{L(t-1)+1} \cdots y_{Lt}$$

and produces as output

$$\mathsf{Expand}(\mathbf{y}) = \mathsf{DeMul}(y_1 \cdots y_L) \| \cdots \| \mathsf{DeMul}(y_{L(t-1)+1} \cdots y_{Lt})$$

where $\mathsf{DeMul} : \{0,1\}^L \to \{0,1\}^{2^L}$ is a demultiplexer function that on input $z \in \{0,1\}^L$ outputs a 2^L-bit string which is 1 in exactly the z-th location (and 0 everywhere else). Then, we have that

1. *(Expand **has sparse outputs**). for all $\mathbf{y} \in \{0,1\}^k$: $|\mathsf{Expand}(\mathbf{y})| = t$;*
2. *(Expand **is injective**). Expand is injection with $k = L \cdot t = \log(q/t)t$.*
3. *(Expand **is parallelizable**). Expand is contained in AC^0.*

Our framework is based on the following expand-then-compress construction.

Construction 3.1 *Let $k = k(n)$ and $q = q(n)$ be integer valued functions, and let $\mathsf{Expand} : \{0,1\}^k \to \{0,1\}^q$ be an expanding function as in Lemma 3.1. A collection of functions $\mathcal{H}_{k,n} = \{h_{\mathbf{M}} : \{0,1\}^k \to \{0,1\}^n, \mathbf{M} \in \{0,1\}^{n \times q}\}$ is defined as*

$$h_{\mathbf{M}}(\mathbf{x}) = \mathbf{M} \cdot \mathsf{Expand}(\mathbf{x})$$

where the key-sampler $\mathcal{G}(1^n)$ samples an $n \times q$ matrix $\mathbf{M} \xleftarrow{\$} \{0,1\}^{n \times q}$.

3.2 The Main Framework of LPN-based CRH

We state our main framework in Theorem 3.1 and then derive the main feasibility results in Corollary 3.1.

Theorem 3.1 (The main framework). *Let n be the security parameter, and let $\mu = \mu(n)$, $k = k(n)$, $q = q(n)$, $t = t(n)$ and $T = T(n)$ such that $t \leq \sqrt{q}$ and $q \leq T = 2^{\frac{8\mu t}{\ln 2(1-2\mu)}}$. Assume that the (n,μ,q)-DLPN problem is T-hard, and let $h_{\mathbf{M}}$ and Expand be defined as in Lemma 3.1 and Construction 3.1 respectively. Then, for every probabilistic adversary $\mathcal{A}$ of running time $T' = 2^{\frac{4\mu t}{\ln 2(1-2\mu)}-1}$*

$$\Pr_{\mathbf{M} \xleftarrow{\$} \{0,1\}^{n \times q}} [\ (\mathbf{y}, \mathbf{y}') \leftarrow \mathcal{A}(\mathbf{M}) : \mathbf{y} \neq \mathbf{y}' \wedge h_{\mathbf{M}}(\mathbf{y}) = h_{\mathbf{M}}(\mathbf{y}')\ ] \leq \frac{1}{T'} \ .$$

We do not say "$h_{\mathbf{M}}$ is a T'-hard CRH" as it may not be $\mathsf{poly}(n)$-time computable.

Proof. Suppose for contradiction that $\mathcal{A}$ finds out a collision with probability more than $1/T'$ s.t. $\mathbf{y} \neq \mathbf{y}'$ and $h_{\mathbf{M}}(\mathbf{y}) = h_{\mathbf{M}}(\mathbf{y}')$, then we have $\mathbf{M} \cdot \mathbf{x} = \mathbf{0}$, where $\mathbf{x} = \mathsf{Expand}(\mathbf{y}) - \mathsf{Expand}(\mathbf{y}') \neq 0$ due to the distinctiveness of Expand, and

$$|\mathbf{x}| \leq |\mathsf{Expand}(\mathbf{y})| + |\mathsf{Expand}(\mathbf{y}')| \leq 2t \ .$$

We define in Algorithm 1 below an LPN distinguisher $\mathcal{D}$ that on input $(\mathbf{M}^\mathsf{T}, \mathbf{z})$, where $\mathbf{M}^\mathsf{T} \xleftarrow{\$} \{0,1\}^{q \times n}$, and either $\mathbf{z} = \mathbf{M}^\mathsf{T}\mathbf{s} + \mathbf{e}$ (for $\mathbf{e} \leftarrow \mathcal{B}_\mu^q$) or $\mathbf{z} \xleftarrow{\$} \{0,1\}^q$, invokes $\mathcal{A}$ on $\mathbf{M}$, and if a collision $(\mathbf{y}, \mathbf{y}')$ is found, it outputs $\mathbf{x}^\mathsf{T}\mathbf{z}$ for $\mathbf{x} = \mathsf{Expand}(\mathbf{y}) - \mathsf{Expand}(\mathbf{y}')$, and otherwise it outputs a uniform random bit. On a successful collision, we have by Lemma 2.1 and Fact 1

$$\Pr[\mathbf{x}^\mathsf{T}\mathbf{z} = \mathbf{x}^\mathsf{T}\mathbf{e} = 0] \geq \frac{1}{2} + \frac{2^{-(\log \frac{1}{1-2\mu})2t}}{2} \geq \frac{1}{2} + \frac{2^{-\frac{4\mu t}{\ln 2(1-2\mu)}}}{2} \ .$$

Therefore, $\mathcal{D}$ achieves an overall advantage of

$$\Pr[\mathcal{D}(\mathbf{M}^\mathsf{T}, \mathbf{M}^\mathsf{T}\mathbf{s} + \mathbf{e}) = 0] - \Pr_{\mathbf{z} \xleftarrow{\$} \{0,1\}^q}[\mathcal{D}(\mathbf{M}^\mathsf{T}, \mathbf{z}) = 0]$$

$$> \frac{1}{T'} \cdot \frac{2^{-\frac{4\mu t}{\ln 2(1-2\mu)}}}{2} \geq 2^{-\frac{8\mu t}{\ln 2(1-2\mu)}} \ ,$$

which is a contradiction to the assumption.

Algorithm 1 A distinguisher $\mathcal{D}$ for (n,μ,q)-DLPN

Input: $(\mathbf{M}^\mathsf{T}, \mathbf{z})$, where $\mathbf{M}^\mathsf{T} \in \{0,1\}^{q \times n}$ and $\mathbf{z} \in \{0,1\}^q$
$(\mathbf{y}, \mathbf{y}') \leftarrow \mathcal{A}(\mathbf{M})$;
$\mathbf{x} = \mathbf{y} - \mathbf{y}'$;
if $0 < |\mathbf{x}| \leq 2t \wedge \mathbf{Mx} = \mathbf{0}$ **then**
  $v = \mathbf{x}^\mathsf{T}\mathbf{z}$
else
  $v \xleftarrow{\$} \{0,1\}$
end if
Output: v

Corollary 3.1 (Main feasibility results). *Assume that (n,μ,q)-DLPN is T-hard, then T'-hard CRH functions with shrinkage $\frac{n}{k}$ exist in $\mathsf{AC}^0(\mathsf{MOD2})$, where $(\mu,T,q,T',\frac{n}{k})$ can be any of the following:*

1. *(Constant-noise, less hardness, poly-logarithmic shrinkage).* $\mu = O(1)$, $T = 2^{n^{0.5+\varepsilon}}$, $q = 2^{n^{0.5}}$, $T' \approx 2^{n^{0.5+\varepsilon}/2} = \lambda^{\log^{2\varepsilon}\lambda/2}$ *and* $\frac{n}{k} < \frac{16\mu}{\ln 2(1-2\mu)n^\varepsilon} = \frac{16\mu}{\ln 2(1-2\mu)\log^{2\varepsilon}\lambda}$ *for any constant* $\varepsilon > 0$.
2. *(Constant-noise, maximal efficiency, constant shrinkage).* $\mu = O(1)$, $T = 2^{\frac{\varepsilon n}{\log n}}$, $q = n^{C_{\varepsilon,\mu}}$, $T' \approx 2^{\frac{\varepsilon n}{2\log n}}$, $\frac{n}{k} < \frac{1}{2}$ *for any constant* $\varepsilon > 0$ *and* $C_{\varepsilon,\mu} = \max(\frac{32\mu}{\varepsilon \ln 2(1-\mu)}, 2)$.

3. *(Low-noise, maximal efficiency, constant shrinkage).*
   $\mu = 1/\sqrt{n}$, $T = 2^{\frac{\varepsilon\sqrt{n}}{\log n}}$, $q = n^{C_{\varepsilon,\mu}}$, $T' \approx 2^{\frac{\varepsilon\sqrt{n}}{2\log n}}$, $\frac{n}{k} < \frac{1}{2}$ *for any constant* $\varepsilon > 0$ *and* $C'_{\varepsilon,\mu} = \max(\frac{32}{\varepsilon \ln 2}, 2)$.
4. *(Extremely-low-noise, standard hardness, constant shrinkage).*
   $\mu = \frac{(\log n)^2}{n}$, $T = q > \mathsf{poly}(n)$ *and* $T' > \mathsf{poly}(n)$ *for every* poly, *and* $\frac{n}{k} < \frac{1}{2}$.

Proof. Recall that $T = 2^{\frac{8\mu t}{\ln 2(1-2\mu)}}$ and $\frac{n}{k} = \frac{n}{\log(q/t)t} < \frac{2n}{t \log q}$. To prove param #1, we let $\frac{8\mu t}{\ln 2(1-2\mu)} = n^{0.5+\varepsilon}$ and thus $t = \ln 2(1-2\mu)n^{0.5+\varepsilon}/8\mu$, and then with $q = 2^{\sqrt{n}}$ we get

$$\frac{n}{k} < \frac{2\sqrt{n}}{t} < \frac{16\mu}{\ln 2(1-2\mu)n^{\varepsilon}} \ .$$

However, $h_{\mathbf{M}}$ that corresponds to param #2 is not computable in $\mathsf{poly}(n)$, and we need to switch to security parameter $\lambda = 2^{\sqrt{n}}$ s.t. $n = \log^2 \lambda$, $k = \Omega(n^{1+\varepsilon}) = \Omega(\log^{2+2\varepsilon} \lambda)$, $T' = \lambda^{\log^{2\varepsilon} \lambda/2}$. The resulting $h_{\mathbf{M}} : \{0,1\}^{\Omega(\log^{2+2\varepsilon} \lambda)} \to \{0,1\}^{\log^2 \lambda}$ is a T'-hard CRH function on security parameter λ but only operates on small inputs and outputs, and we use parallel repetition (Lemma 3.2) to get a domain/range-extended CRH $h'_{\mathbf{M}} : \{0,1\}^{\Omega(\lambda \log^{2\varepsilon} \lambda)} \to \{0,1\}^{\lambda}$ for $\mathbf{M} \in \{0,1\}^{\log^2 \lambda \times \lambda}$, which is T'-hard and is computable in time $\mathsf{poly}(\lambda)$. Now proceed to params #2 and #3: set $\frac{8\mu t}{\ln 2(1-2\mu)}$ to $\frac{\varepsilon n}{\log n}$ for $\mu = O(1)$ or to $\frac{\varepsilon\sqrt{n}}{\log n}$ for $\mu = 1/\sqrt{n}$, and let $\frac{2n}{t \log q} = \frac{1}{2}$ so that

$$t = \tfrac{\ln 2 \cdot \varepsilon(1-2\mu)n}{8\mu \log n}, \quad \log q = \frac{32\mu \log n}{\varepsilon \ln 2(1-2\mu)} \ \text{ for } \mu = O(1) \ ;$$
$$t = \tfrac{\ln 2 \cdot \varepsilon \cdot n}{8 \log n}, \qquad \log q = \frac{32 \log n}{\varepsilon \ln 2} \ \text{ for } \mu = 1/\sqrt{n} \ .$$

Note that we also need $q \geq n^2$ in respect of the $t \leq \sqrt{q}$ condition. Finally, param #4 is seen by the following: for $\mu = \frac{(\log n)^2}{n}$, any $q = \mathsf{poly}(n)$ and t satisfying $\frac{2n}{t \log q} = 1/2$ we have that $T = 2^{\frac{8\mu t}{\ln 2(1-2\mu)}}$ is another polynomial in n.

Lemma 3.2 (Parallel repetitions of CRH). *Let* $k = k(\lambda)$, $d = d(\lambda)$ *and* $T = T(\lambda)$ *be integer valued functions. If* $\mathcal{H}_{k,\lambda} = \{h_{\mathbf{s}} : \{0,1\}^k \to \{0,1\}^{\lambda}, \mathbf{s} \in \{0,1\}^{\mathsf{poly}(\lambda)}\}$ *is a* T*-hard CRH function, then* $\mathcal{H}'_{dk,d\lambda} = \{h'_{\mathbf{s}} : \{0,1\}^{dk} \to \{0,1\}^{d\lambda}, \mathbf{s} \in \{0,1\}^{\mathsf{poly}(\lambda)}\}$, *where*

$$h'_{\mathbf{s}}(\mathbf{y}_1, \cdots, \mathbf{y}_d) = \big(h_{\mathbf{s}}(\mathbf{y}_1), \cdots, h_{\mathbf{s}}(\mathbf{y}_d)\big), \ \ \mathbf{y}_1, \cdots, \mathbf{y}_d \in \{0,1\}^k \ ,$$

is a (T/d)*-hard CRH function.*

3.3 Assume Less, Shrink More and in Parallel at the Same Time

Although already assuming much less hardness than previously known, the CRH immediately implied by constant-noise LPN (as specified by param #1 of Corollary 3.1) is inefficient as $\mathbf{M}$ is of dimension $n \times 2^{\sqrt{n}}$ and thus the resulting hash

function has computation time far beyond polynomial. The solution by switching to another parameter $\lambda = 2^{\sqrt{n}}$ makes the hash function computable in time polynomial in λ but at the same time it dramatically downgrades the security from $2^{\Omega(n^{1/2+\epsilon})}$ to $\lambda^{\Omega(\log^{2\epsilon}\lambda)}$, and deteriorates the shrinkage factor from polynomial to poly-logarithmic. Otherwise said, we mainly establish feasibility results about basing CRH on constant-noise LPN with minimal hardness possible. We discuss an alternative to void the loss, i.e., to preserve security, polynomial shrinkage and efficiency at the same time. This relies on (arguably minimal) idealized assumptions (e.g., a block cipher keyed with a random public string behaves like a random permutation) in addition to constant-noise LPN. Unlike the parameter-switching technique, this approach applies also to low-noise LPN with even reduced hardness.

Corollary 3.2 (Feasibility results with idealized heuristics). *Let n be the security parameter, and let $\mu = \mu(n)$, $k = k(n)$, $q = q(n)$, $t = t(n)$ and $T = T(n)$ such that $t \leq \sqrt{q}$ and $q \leq T = 2^{\frac{8\mu t}{\ln 2(1-2\mu)}}$. Assume that (n,μ,q)-DLPN is T-hard and $R : \{0,1\}^{\log(q)} \to \{0,1\}^n$ behaves like a random function , then for $\mathbf{y} = \mathbf{y}_1 \| \cdots \| \mathbf{y}_t$ parsed as $t = k/L$ blocks, each of size $L = \log(q/t)$, we have*

$$h^R : \{0,1\}^k \to \{0,1\}^n, \;\; h^R(\mathbf{y}) = \bigoplus_{i=1}^{t} R(i \| \mathbf{y}_i),$$

is a T'-hard CRH function with shrinkage n/k, where $(\mu,T,q,T',\frac{n}{k})$ can be either of the following:

1. *(Constant-noise, less hardness, polynomial shrinkage).*
   $\mu = O(1)$, $T = 2^{n^{0.5+\varepsilon}}$, $q = 2^{n^{0.5}}$, $T' \approx 2^{n^{0.5+\varepsilon}/2}$ and $\frac{n}{k} < \frac{16\mu}{\ln 2(1-2\mu)n^{\varepsilon}}$ for any constant $\varepsilon > 0$.
2. *(Low-noise, less hardness, polynomial shrinkage).*
   $\mu = 1/\sqrt{n}$, $T = 2^{n^{0.25+\varepsilon}}$, $q = 2^{n^{0.25}}$, $T' \approx 2^{n^{0.25+\varepsilon}/2}$ and $\frac{n}{k} < \frac{16}{\ln 2 \cdot n^{\varepsilon}}$ for any constant $\varepsilon > 0$.

Proof. First, assume that h^R is functionally equivalent to $h_{\mathbf{M}}$. Then, param #1 is the same as the counterpart in Corollary 3.1 but we refrain from switching to a new security parameter. To prove param #2, we recall that $T = 2^{\frac{8\mu t}{\ln 2(1-2\mu)}}$ and $\frac{n}{k} = \frac{n}{\log(q/t)t} < \frac{2n}{t \log q}$. Let $\frac{8\mu t}{\ln 2(1-2\mu)} = n^{0.25+\varepsilon}$ and thus $t \approx \ln 2 \cdot n^{0.75+\varepsilon}/8$, and then with $q = 2^{n^{0.25}}$ we get

$$\frac{n}{k} < \frac{2n^{0.75}}{t} < \frac{16}{\ln 2 \cdot n^{\varepsilon}} \;\; .$$

The conclusion then follows from Lemma 3.3 that h^R perfectly instantiates $h_{\mathbf{M}}$.

THE INTUITION. We recall that $h_{\mathbf{M}}(\mathbf{y}) = \mathbf{M} \cdot \mathsf{Expand}(\mathbf{y})$ for an $n \times q$ matrix $\mathbf{M}$ and that Expand parses $\mathbf{y}$ into $t = k/L$ blocks and produces same number of output blocks accordingly, where $q = t \cdot 2^L$. We also parse $\mathbf{M}$ into t equal-size

submatrices $\mathbf{M}_1, \cdots, \mathbf{M}_t$, each of dimension $n \times 2^L$. Let $R : \{0,1\}^{\log(q)} \to \{0,1\}^n$ be a random function that describes $\mathbf{M}$, i.e., for every $j \in \{0,1\}^{\log(q)}$ the output $R(j)$ corresponds to the j-th column of $\mathbf{M}$. We thus have

$$h_{\mathbf{M}}(\mathbf{y}) = \underbrace{\left[\mathbf{M}_1 \cdots \mathbf{M}_t\right]}_{\mathbf{M}} \cdot \underbrace{\begin{bmatrix} \mathsf{DeMul}(\mathbf{y}_1) \\ \vdots \\ \mathsf{DeMul}(\mathbf{y}_t) \end{bmatrix}}_{\mathsf{Expand}(\mathbf{y})} = \bigoplus_{i=1}^{t} R(i \| \mathbf{y}_i) \tag{5}$$

where $R(i\|\mathbf{y}_i) = \mathbf{M}_i \cdot \mathsf{DeMul}(\mathbf{y}_i)$ simply follows the definition of R and DeMul. Therefore, the task of constructing polynomially shrinking CRH functions from constant-noise LPN is now reduced to instantiating a small-domain random function $R : \{0,1\}^{\log(q)} \to \{0,1\}^n$ for $\log(q) \ll n$.

Lemma 3.3 (An idealized realization of $h_{\mathbf{M}}$). *Let $k = k(n)$, $t = t(n)$, $q = q(n)$ and $L = L(n)$ be integer valued functions such that $q/t = 2^L$. Assume that $R : \{0,1\}^{\log(q)} \to \{0,1\}^n$ behaves like a random function, then $h^R(\mathbf{y}) = \bigoplus_{i=1}^{t} R(i\|\mathbf{y}_i)$ as defined in (5) perfectly realizes $h_{\mathbf{M}}$ specified in Construction 3.1*

One may want to replace R with a pseudorandom function (with key made public), but in general the security cannot be achieved with a standard reducibility argument due to the distinction between public-coin and secret-coin CRH functions [37]. We thus resort to random permutations or idealized blockciphers.

Random functions vs. permutations. The small-domain random function (to be instantiated) is not commonly found in practice, but it is implied by a large-domain random function for free, i.e., $R(x) = F(0^l\|x)$ is a random function if F is a random one. Thus, we simply consider a length-preserving random function, which can be in turn based on a random permutation (and instantiated with block ciphers). For example, for random permutations π, π_1, π_2, we have that $\pi \oplus \pi^{-1}$ [28] (or $\pi_1 \oplus \pi_2$ [52]) is indifferentiable from a length-preserving random function. This means that R on input x can be instantiated as

$$\mathsf{AES}_k(0^l\|x) \oplus \mathsf{AES}_k^{-1}(0^l\|x) \text{ or } \mathsf{AES}_{k_1}(0^l\|x) \oplus \mathsf{AES}_{k_2}(0^l\|x)$$

where $l = n - \log(q)$ bits are padded to fit into a permutation, k, k_1, and k_2 are public random keys. Intuitively, the XOR of a permutation and its inverse (or two independent permutations) is to destroy the permutation structure as its inversibility could give the adversary additional advantages in collision finding. The former instantiation relies on the assumption that a practical block cipher like AES on a random key behaves like a random permutation. We reproduce below the results by Dodis et al. [28] that $\pi \oplus \pi^{-1}$ is indifferentiable from a (length-preserving) random function. Therefore, instantiation of a random function with a blockcipher only incurs a factor of 2 in the number of calls to the underlying primitive.

Lemma 3.4 (Lemma 4 from [28]). *Let n be the security parameter, let $q = q(n)$ and let π be a random permutation over $\{0,1\}^n$. We have that $\pi \oplus \pi^{-1}$ is $(q, q, O(nq), O(\frac{q^2}{2^n}))$-indifferentiable from an n-to-n-bit random function.*

On related works. We offer a new construction of CRH functions from *fixed-key block ciphers/random permutations*. Compared with the traditional blockcipher-based compression functions, e.g. [54,56,13], our solution avoids the key-setup costs and eliminates the need for related-key security on a large space of keys. That is, (using AES-128 as an example) we only assume that "AES on a single random key behaves like a random permutation", instead of that "AES on 2^{128} keys yields 2^{128} independent random permutations", as imposed by the Ideal Cipher Model. On the other hand, existing permutation-based solutions either only offer a constant shrinkage factor (typically 1/2) [61,51], or require permutations with a large domain (e.g., [29] needs a large permutation on n^2-bit strings to obtain a CRH function with shrinkage factor $1/n$), and in contrast our construction runs in parallel and compresses polynomially.

4 Concluding Remarks

We present a general framework for constructing CRH from LPN for a broad spectrum of parameter choices, and thus resolve the open problem whether CRH functions can be based on the (reasonable) hardness of LPN. We also discuss how to improve the efficiency using idealized heuristics.

References

1. Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing (STOC 1996). pp. 99–108 (1996)
2. Alekhnovich, M.: More on average case vs approximation complexity. In: 44th Annual Symposium on Foundations of Computer Science. pp. 298–307. IEEE, Cambridge, Massachusetts (Oct 2003)
3. Andreeva, E., Mennink, B., Preneel, B.: Security properties of domain extenders for cryptographic hash functions. Journal of Information Processing Systems 6(4), 453–480 (2010), http://dx.doi.org/10.3745/JIPS.2010.6.4.453
4. Applebaum, B., Cash, D., Peikert, C., Sahai, A.: Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In: Advances in Cryptology - CRYPTO 2009. pp. 595–618 (2009)
5. Applebaum, B., Haramaty, N., Ishai, Y., Kushilevitz, E., Vaikuntanathan, V.: Low-complexity cryptographic hash functions. In: Proceedings of the 2017 Conference on Innovations in Theoretical Computer Science (ITCS 2017). pp. ??–?? (2017)
6. Applebaum, B., Ishai, Y., Kushilevitz, E.: Cryptography with constant input locality. In: Advances in Cryptology - CRYPTO 2007. pp. 92–110 (2007), full version available at http://www.eng.tau.ac.il/~bennyap/pubs/input-locality-full-revised-1.pdf
7. Arbitman, Y., Dogon, G., Lyubashevsky, V., Micciancio, D., Peikert, C., Rosen, A.: SWIFFTX: A proposal for the sha-3 standard (2009), http://www.eecs.harvard.edu/~alon/PAPERS/lattices/swifftx.pdf
8. Augot, D., Finiasz, M., Gaborit, P., Manuel, S., Sendrier, N.: Sha-3 proposal: Fsb (2008), https://www.rocq.inria.fr/secret/CBCrypto/fsbdoc.pdf

9. Becker, A., Joux, A., May, A., Meurer, A.: Decoding random binary linear codes in $2^{n/20}$: How $1 + 1 = 0$ improves information set decoding. In: Advances in Cryptology - EUROCRYPT 2012. pp. 520–536 (2012)
10. Berlekamp, E., McEliece, R.J., van Tilborg, H.: On the inherent intractability of certain coding problems. IEEE Transactions on Information Theory 24(3), 384–386 (1978)
11. Bernstein, D.J., Lange, T., Peters, C.: Smaller decoding exponents: Ball-collision decoding. In: Advances in Cryptology - CRYPTO 2011. pp. 743–760 (2011)
12. Bernstein, D.J., Lange, T., Peters, C., Schwabe, P.: Really fast syndrome-based hashing. In: Progress in Cryptology - AFRICACRYPT 2011. pp. 134–152 (2011)
13. Black, J., Rogaway, P., Shrimpton, T., Stam, M.: An analysis of the blockcipher-based hash functions from pgv. Journal of Cryptology 23(4), 519–545 (2010)
14. Blum, A., Furst, M.L., Kearns, M.J., Lipton, R.J.: Cryptographic primitives based on hard learning problems. In: Stinson, D.R. (ed.) Advances in Cryptology—CRYPTO '93. LNCS, vol. 773, pp. 278–291. Springer-Verlag (22–26 Aug 1993)
15. Blum, A., Kalai, A., Wasserman, H.: Noise-tolerant learning, the parity problem, and the statistical query model. Journal of the ACM 50(4), 506–519 (2003)
16. Brakerski, Z.: When homomorphism becomes a liability. In: Proceedings of 10th Theory of Cryptography Conference (TCC 2013). pp. 143–161 (2013)
17. Brakerski, Z., Lombardi, A., Segev, G., Vaikuntanathan, V.: Anonymous ibe, leakage resilience and circular security from new assumptions. Cryptology ePrint Archive, Report 2017/967 (2017), https://eprint.iacr.org/2017/967
18. Brakerski, Z., Lyubashevsky, V., Vaikuntanathan, V., Wichs, D.: LPN and other noisy decoding problems (working title). not publicly available as of Dec 29 2017 other than being cited by ePrint/2017/967 (2017)
19. Canteaut, A., Chabaud, F.: A new algorithm for finding minimum-weight words in a linear code: Application to mceliece's cryptosystem and to narrow-sense BCH codes of length 511. IEEE Transactions on Information Theory 44(1), 367–378 (1998)
20. Cash, D., Kiltz, E., Tessaro, S.: Two-round man-in-the-middle security from LPN. In: Proceedings of the 13th Theory of Cryptography (TCC 2016-A). pp. 225–248 (2016)
21. Coron, J.S., Dodis, Y., Malinaud, C., Puniya, P.: Merkle-damgård revisited: How to construct a hash function. In: Shoup [60], pp. 430–448
22. Damgård, I.: Collision free hash functions and public key signature schemes. In: Advances in Cryptology - EUROCRYPT '87. pp. 203–216 (1987)
23. Damgård, I.: A design principle for hash functions. In: Advances in Cryptology - CRYPTO '89. pp. 416–427 (1989)
24. David, B., Dowsley, R., Nascimento, A.C.A.: Universally composable oblivious transfer based on a variant of LPN. In: Proceedings of the 13th International Conference on Cryptology and Network Security (CANS 2014). pp. 143–158 (2014)
25. Demay, G., Gaži, P., Hirt, M., Maurer, U.: Resource-Restricted Indifferentiability. In: Johansson, T., Nguyen, P.Q. (eds.) Advances in Cryptology – EUROCRYPT 2013, Lecture Notes in Computer Science, vol. 7881, pp. 664–683. Springer Berlin Heidelberg (2013)
26. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Transactions on Information Theory IT-22(6), 644–654 (1976)
27. Dodis, Y., Kiltz, E., Pietrzak, K., Wichs, D.: Message authentication, revisited. In: Proceedings of the 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2012). pp. 355–374 (2012)

28. Dodis, Y., Pietrzak, K., Puniya, P.: A new mode of operation for block ciphers and length-preserving macs. In: Smart, N.P. (ed.) Advances in Cryptology - EUROCRYPT 2008. LNCS, vol. 4965, pp. 198–219. Springer-Verlag (2008)
29. Dodis, Y., Reyzin, L., Rivest, R.L., Shen, E.: Indifferentiability of Permutation-Based Compression Functions and Tree-Based Modes of Operation, with Applications to MD6. In: Dunkelman, O. (ed.) 15th International Workshop Fast Software Encryption (FSE 2009). Lecture Notes in Computer Science, vol. 5665, pp. 104–121. Springer Berlin Heidelberg (2009)
30. Döttling, N.: Low noise lpn: Kdm secure public key encryption and sample amplification. In: Public-Key Cryptography–PKC 2015. pp. 604–626. Springer (2015)
31. Döttling, N., Müller-Quade, J., Nascimento, A.C.A.: IND-CCA secure cryptography based on a variant of the LPN problem. In: Advances in Cryptology – ASIACRYPT 2012. pp. 485–503 (2012)
32. Dumer, I., Micciancio, D., Sudan, M.: Hardness of approximating the minimum distance of a linear code. IEEE Transactions on Information Theory 49(1), 22–37 (2003)
33. Esser, A., Kübler, R., May, A.: LPN decoded. In: Advances in Cryptology - CRYPTO 2017, Part II. pp. 486–514 (2017)
34. Feldman, V., Gopalan, P., Khot, S., Ponnuswami, A.K.: New results for learning noisy parities and halfspaces. In: 47th Symposium on Foundations of Computer Science. pp. 563–574. IEEE, Berkeley, CA, USA (Oct 21–24 2006)
35. Gertner, Y., Kannan, S., Malkin, T., Reingold, O., Viswanathan, M.: The relationship between public key encryption and oblivious transfer. In: Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on. pp. 325–335 (2000)
36. Hopper, N.J., Blum, M.: Secure human identification protocols. In: Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT 2001). pp. 52–66 (2001)
37. Hsiao, C.Y., Reyzin, L.: Finding collisions on a public road, or do secure hash functions need secret coins. In: Franklin, M. (ed.) Advances in Cryptology—CRYPTO 2004. LNCS, vol. 3152, pp. 92–105. Springer-Verlag (15–19 Aug 2004)
38. Jain, A., Krenn, S., Pietrzak, K., Tentes, A.: Commitments and efficient zero-knowledge proofs from learning parity with noise. In: Advances in Cryptology – ASIACRYPT 2012. pp. 663–680 (2012)
39. Juels, A., Weis, S.A.: Authenticating pervasive devices with human protocols. In: Shoup [60], pp. 293–308
40. Katz, J., Shin, J.S.: Parallel and concurrent security of the hb and hb$^+$ protocols. In: Vaudenay, S. (ed.) Advances in Cryptology—EUROCRYPT 2006. LNCS, vol. 4004, pp. 73–87. Springer-Verlag (2006)
41. Kiltz, E., Masny, D., Pietrzak, K.: Simple chosen-ciphertext security from low-noise lpn. In: Proceeding of the 17th Conference on Theory and Practice in Public Key Cryptography (PKC 2014), pp. 1–18 (2003)
42. Kiltz, E., Pietrzak, K., Cash, D., Jain, A., Venturi, D.: Efficient authentication from hard learning problems. In: Proceedings of the 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2011). pp. 7–26 (2011)
43. Kirchner, P.: Improved generalized birthday attack. Cryptology ePrint Archive, Report 2011/377 (2011), http://eprint.iacr.org/2011/377
44. Kirchner, P., Fouque, P.: An improved BKW algorithm for LWE with applications to cryptography and lattices. In: Advances in Cryptology - CRYPTO 2015 - 35th

Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I. pp. 43–62 (2015), full version at https://eprint.iacr.org/2015/552.pdf
45. Lyubashevsky, V.: The parity problem in the presence of noise, decoding random linear codes, and the subset sum problem. In: Proceedings of the 9th International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM 2005). pp. 378–389 (2005)
46. Lyubashevsky, V.: The LPN problem in cryptography. Invited talk at the 14th IMA International ConferenceO On Cryptography and Coding (2013), refer to http://www.cs.bris.ac.uk/Research/CryptographySecurity/IMACC13/Web/programme.html for the programme and goo.gl/zpHFp7 for the slides.
47. Lyubashevsky, V., Masny, D.: Man-in-the-middle secure authentication schemes from lpn and weak prfs. In: Advances in Cryptology - CRYPTO 2013. pp. 308–325 (2013)
48. Lyubashevsky, V., Micciancio, D., Peikert, C., Rosen, A.: SWIFFT: A modest proposal for FFT hashing. In: 15th International Workshop Fast Software Encryption (FSE 2008). pp. 54–72 (2008)
49. Maurer, U., Renner, R., Holenstein, C.: Indifferentiability, Impossibility Results on Reductions, and Applications to the Random Oracle Methodology. In: Naor, M. (ed.) Theory of Cryptography, Lecture Notes in Computer Science, vol. 2951, pp. 21–39. Springer Berlin Heidelberg (2004)
50. May, A., Meurer, A., Thomae, E.: Decoding random linear codes in $\tilde{O}(2^{0.054n})$. In: Proceedings of the 17th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT 2011). pp. 107–124 (2011)
51. Mennink, B., Preneel, B.: Hash Functions Based on Three Permutations: A Generic Security Analysis. In: Safavi-Naini, R., Canetti, R. (eds.) Advances in Cryptology – CRYPTO 2012. Lecture Notes in Computer Science, vol. 7417, pp. 330–347. Springer Berlin Heidelberg (2012)
52. Mennink, B., Preneel, B.: On the XOR of Multiple Random Permutations. In: Malkin, T., Kolesnikov, V., Lewko, A.B., Polychronakis, M. (eds.) Applied Cryptography and Network Security: 13th International Conference, ACNS 2015. Lecture Notes in Computer Science, vol. 9092, pp. 619–634. Springer Berlin Heidelberg (2015)
53. Merkle, R.: Secrecy, Authentication, and Public Key Systems. Ph.D. thesis (1979)
54. Merkle, R.C.: One way hash functions and DES. In: Advances in Cryptology - CRYPTO '89. pp. 428–446 (1989)
55. Pietrzak, K.: Cryptography from learning parity with noise. In: Proceedings of the Theory and Practice of Computer Science (SOFTSEM 2012). pp. 99–114 (2012)
56. Preneel, B., Govaerts, R., Vandewalle, J.: Hash Functions Based on Block Ciphers: A Synthetic Approach. In: Stinson, D.R. (ed.) Advances in Cryptology - CRYPTO' 93. Lecture Notes in Computer Science, vol. 773, pp. 368–378. Springer Berlin Heidelberg (1994)
57. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Gabow, H.N., Fagin, R. (eds.) STOC. pp. 84–93. ACM (2005)
58. Ristenpart, T., Shacham, H., Shrimpton, T.: Careful with Composition: Limitations of the Indifferentiability Framework. In: Paterson, K. (ed.) Advances in Cryptology – EUROCRYPT 2011, Lecture Notes in Computer Science, vol. 6632, pp. 487–506. Springer Berlin Heidelberg (2011)
59. Rogaway, P., Shrimpton, T.: Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and

collision resistance. In: 11th International Workshop on Fast Software Encryption (FSE 2004). pp. 371–388 (2004)
60. Shoup, V. (ed.): Advances in Cryptology—CRYPTO 2005, LNCS, vol. 3621. Springer-Verlag (14–18 Aug 2005)
61. Shrimpton, T., Stam, M.: Building a Collision-Resistant Compression Function from Non-compressing Primitives (Extended Abstract). In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) Automata, Languages and Programming – ICALP 2008, Part II, Lecture Notes in Computer Science, vol. 5126, pp. 643–654. Springer Berlin Heidelberg (2008)
62. Vardy, A.: The intractability of computing the minimum distance of a code. IEEE Transactions on Information Theory 43(6), 1757–1766 (1997)
63. Yu, Y., Zhang, J.: Cryptography with auxiliary input and trapdoor from constant-noise lpn. In: Advances in Cryptology – CRYPTO 2016. pp. 214–243 (2016)