Learning Parity with Noise Implies Collision Resistant Hashing

Yu Yu[1,2], Jiang Zhang[2], Jian Weng[3], Chun Guo[1], and Xiangxue Li[4]

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China
[3] Jinan University, Guangzhou, China
[4] East China Normal University, Shanghai,China
E-mail: {yuyuathk, jiangzhang09}@gmail.com

Abstract. The Learning Parity with Noise (LPN) problem has recently found many cryptographic applications such as authentication protocols, pseudorandom generators/functions and even cryptomania 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. In this paper, we answer this question affirmatively by showing that CRH is implied by (the two most common variants of) LPN. More specifically, for any constant $\epsilon > 0$, assume that

1. the low-noise LPN problem (i.e., at noise rate $1/\sqrt{n}$) is $2^{4\sqrt{n}/\log n}$-hard given $q = n^{3+\epsilon}$ samples,
2. or that the constant-noise LPN problem is $2^{n^{0.5+\epsilon}}$-hard,

then there exists CRH functions with constant (resp., 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.

In addition to the feasibility established, we discuss also the practical relevance of the CRH functions constructed (from the hardness of LPN). Interestingly, the SHA-3 proposal Fast Syndrome Based (FSB) hash resembles a concrete (but aggressive) instantiation of the LPN-based CRH construction. Furthermore, we show how to implement the polynomially shrinking CRH functions more efficiently using idealized heuristics such as a block cipher (keyed by a public random string) behaves like a random permutation.

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

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 (i.e., $\Pr[\mathcal{B}_\mu = 1] = \mu$ and $\Pr[\mathcal{B}_\mu = 0] = 1 - \mu$), '$\cdot$' 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 [16,47,6].

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 [35] 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 [17] showed how to solve LPN with time/sample complexity $2^{O(n/\log n)}$. Lyubashevsky [51] 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 [17] 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 $2^{O(n)}$ [55]. Under low noise rate $\mu = 1/\sqrt{n}$, the best attacks [21,13,50,9] 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 are not known to have any advantages over classic ones in solving LPN, which 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) [62] as the many modular additions and multiplications in LWE would be simplified to AND and XOR gates in LPN.

Cryptography in minicrypt[5]. LPN was used as a basis for building lightweight authentication schemes (e.g. [42,46,47], just to name a few). Kiltz et al. [49] and Dodis et al. [29] constructed randomized MACs from LPN, which implies a two-round authentication scheme with security against active adversaries. Lyubashevsky and Masny [52] gave a more efficient three-round authentication scheme

[5] minicrypt refers to Impagliazzo's [44] hypothetical world where one-way functions exist but public-key cryptography does not, and cryptomania is the more optimistic world where public-key cryptography and multiparty computation are possible.

from LPN and recently Cash, Kiltz, and Tessaro [22] 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. [45] constructed an efficient perfectly binding string commitment scheme from LPN. We refer to the survey [60] about cryptography from LPN.

Cryptography beyond minicrypt. Alekhnovich [2] constructed the first cryptomania object from LPN, in particular, he showed that LPN with noise rate $1/\sqrt{n}$ implies CPA secure public-key encryption (PKE) schemes. Döttling et al. [33] and Kiltz et al. [48] further showed that low-noise LPN alone already suffices for PKE schemes with CCA (and KDM [32]) 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. [38] showed how to construct an OT protocol in a black-box manner. This observation was made explicit in [26], where universally composable OT protocols were constructed from low-noise LPN. All the above schemes are based on LPN of noise rate $1/\sqrt{n}$. Recently, Yu and Zhang [69] showed that PKE and OT can be based on constant-noise LPN with hardness $2^{n^{1/2+\epsilon}}$. To summarize, it remains open whether LPN implies more advanced cryptograpic objects, such as fully homomorphic encryption (FHE) and collision resistant hash (CRH) functions [60]. As for LPN-based FHE, Brakerski [18] reported some negative result that straightforward LPN-based encryptions are unlikely to achieve full homomorphism. We tackle the case of LPN-based CRH.

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 polynomially) long bit strings into digests of a fixed length. The function was originally introduced in the seminal work of Diffie and Hellman [28] 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 [58] 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 [6]. 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

[6] 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 [64].

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 [59] and Damgård [25], 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, termed collision resistant hash (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 [24]), 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. [53] gave a ring-based variant of Ajtai's construction, called SWIFFT, which admits FFT and precomputation techniques for improved efficiency and at the same time it preserves an asymptotic security proof from ideal lattices. Despite a big gap between the claimed security level and the actual security bounds proved, SWIFFT [53] 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).

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 $n \times \alpha n$ 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 a k-bit string into an αn-bit one of weight no greater than $\delta \alpha n/2$ for some small $\delta < 1$. Note that $h_{\mathbf{M}}$ can be viewed as a binary version of Ajtai's $f_{\mathbf{A}}$ (see (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 $\mathsf{Expand}(\mathbf{y})$. As suggested in [5] (and already used in the FSB hash [8]), $\mathsf{Expand} : \{0,1\}^k \to \{0,1\}^{\alpha n}$ can be implemented efficiently and in parallel as follows: assume that δ is a negative power of 2 and $L|k$ (i.e., $L = \log(2/\delta), k/L \in \mathbb{N}$) and let $\alpha = k2^L/Ln$, then Expand parses the k-bit input into L-bit blocks as

$$\mathbf{y} = y_1 \cdots y_L \| y_{L+1} \cdots y_{2L} \| \cdots \| y_{L(k/L-1)+1} \cdots y_k$$

and produces as output

$$\mathsf{Expand}(\mathbf{y}) = \mathsf{DeMul}(y_1 \cdots y_L) \| \cdots \| \mathsf{DeMul}(y_{L(\frac{k}{L}-1)+1} \cdots y_k) \ ,$$

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 elsewhere). This yields an output of length $\alpha n = k2^L/L$ and Hamming weight $k/L = \delta\alpha n/2$. Thanks to the simplification to the binary field, $h_{\mathbf{M}}$ can be implemented rather efficiently both in the asymptotic sense and in practice. For any specified $\mathbf{M}$, $h_{\mathbf{M}}$ 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). The original FSB hash proposal [8] and its improved version the RFSB hash [14] can be seen as concrete instantiations of the Expand-then-Compress hash (although they appeared earlier than [5]). The FSB and RFSB hash functions were shown to be only 8 times slower or even faster than SHA-256 respectively (see the evaluations in [11,14]).

The security of EtC and Binary SVP. An early version of FSB [36] for some concrete parameter settings was broken by Saarinen [65], who showed that using too many blocks (large k/L compared with n) may cause linearization attacks and suggested adjusted parameter choices. Another attack by Fouque and Leurent [37] targets at a quasi-cyclic version of FSB, where a quasi-cyclic (instead of random) matrix and a wrong parameter choice (violating the assumption) were used in [36], and is thus irrelevant to the EtC methodology. The official version of FSB [8] fixed these problems and remains unbroken to date (except for attacks on its scale-downed version [12]). In order to study the asymptotic security of the EtC hash, Applebaum et al. [5] introduced the binary Shortest Vector Problem (binary SVP or bSVP in short). Informally, the (α, δ)-bSVP assumption asserts that given a random matrix distribution[7] $\mathbf{M} \xleftarrow{\$} \{0,1\}^{n\times\alpha n}$, it is computationally infeasible to find a non-zero $\mathbf{x} \in \{0,1\}^{\alpha n}$ of low Hamming weight $|\mathbf{x}| \le \delta\alpha n$ such that $\mathbf{Mx} = \mathbf{0} \mod 2$. From a code-theoretic perspective, $\mathbf{M}$ specifies the $n \times \alpha n$ parity check matrix of a random binary linear code of rate $1-1/\alpha$, where the rows of $\mathbf{M}$ are linearly independent (except with negligible probability), and therefore the bSVP postulates that finding a short codeword (of weight at most $\delta\alpha n$) is hard. Meaningful regimes (i.e., upper and lower bounds) for δ were discussed in [5]: for $\delta < \mathbf{H}^{-1}(1/\alpha)$ such short codes do not exist (except with exponentially small probability), and for $\delta \ge 1/2\alpha$ the b-SVP assumption is falsified by an efficient attack, where $\mathbf{H}^{-1}(\cdot)$ denotes the inverse of the binary entropy function, i.e., $\mathbf{H}(v) = v \log 1/v + (1-v)\log 1/(1-v)$ for $0 < v \le 1/2$. Therefore, b-SVP could only live in the range of $\delta \in (\mathbf{H}^{-1}(1/\alpha), 1/2\alpha)$, and the authors of [5] further show that one-way functions and collision resistant hash functions exist if the bSVP is hard for $\delta > \mathbf{H}^{-1}(1/\alpha)$ and $\delta > 2\mathbf{H}^{-1}(1/\alpha)$ respectively. To see that collision resistant hash functions are possible for $\delta > 2\mathbf{H}^{-1}(1/\alpha)$, consider $h_{\mathbf{M}}$ as defined in (2). Note that $h_{\mathbf{M}}$ can be made shrinking since $k \approx \log \binom{\alpha n}{\delta\alpha n/2} \approx \alpha n \cdot \mathbf{H}(\delta/2) > n$ (Lemma 4). The rationale is that 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

[7] For our convenience, the matrix $\mathbf{M}$ in our consideration has dimension $n \times \alpha n$ instead of $\alpha n \times n$ in [5]. Thus, we adjust parameters accordingly when citing [5].

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 $\delta\alpha n$. 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 [68,34]. 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 authors of [5] only showed a weak connection between bSVP and LPN. That is, they show that at least one of the following is true:

1. One can achieve an arbitrary polynomial speedup over the BKW algorithm [17], i.e., for every constant $c > 0$ and there exists an algorithm that solves the n-dimensional constant-noise LPN with time and sample complexity $\mathsf{poly}(n) \cdot 2^{\frac{cn}{\log n}}$ for infinitely many n's.
2. There exists $a > 1$ for which (α, δ)-bSVP holds for $\alpha = n^{1/(a-1)}$ and $\delta = 8/(\alpha \log(\alpha))$, which implies CRH functions of constant shrinkage factor and logarithmic degree.

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

On the concurrent work of [20]. To our best knowledge[8], Brakerski et al. [20] gave a construction of CRH from LPN of the extremely low noise rate $\log^2 n/n$, which is thus incomparable to our work.

To summarize, it remains an open problem whether efficient CRH functions exist assuming reasonable hardness of the LPN problem in typical symmetric and asymmetric noise regimes, i.e., $\mu = O(1)$ and $\mu = 1/\sqrt{n}$ respectively, which we will tackle in this paper.

1.3 Our Work

Our contribution. We summarize the contributions as follows. Assume that

- the LPN problem at noise rate $1/\sqrt{n}$ is at least $2^{4\sqrt{n}/\log n}$-hard given $q = n^{3+\epsilon}$ samples,
- or that the LPN problem at any constant noise rate $0 < \mu < 1/2$ is at least $2^{n^{1/2+\epsilon}}$-hard,

for any constant $\epsilon > 0$, then there exists a CRH function with constant (or poly-logarithmic for the latter assumption) shrinkage, which can be constructed following the Expand-then-Compress approach and can be implemented by a polynomial-size depth-3 circuit with NOT, (unbounded fan-in) AND and XOR

[8] [20] hasn't been publicly available yet (by the end of 2017) and the only information we learn about [20] is through [19], where the Sections 1.1 and 2.3 of [19] summarize the CRH construction of [20] assuming LPN with an extremely low noise rate.

gates[9]. We remark that the $2^{\Omega(\sqrt{n}/\log n)}$-hardness assumption for LPN of noise rate $1/\sqrt{n}$ is quite reasonable as the current best attacks need complexity $2^{\Omega(\sqrt{n})}$ [13,50,9], and the $2^{n^{0.501}}$-hardness assumption about constant-noise LPN offers even more generous security margins as the best attack goes even beyond complexity $2^{n^{0.999}}$ [17]. Therefore, we establish the feasibility that collision resistant hash functions can be based on the reasonable hardness of LPN in commonly used noise regimes, which was a longstanding open problem [60]. The construction follows a parallel and conceptually simple approach and resembles a binary version of Ajtai's SIS-based construction, of which the FSB hash and its variants fall into concrete (but over optimistic) instantiations.[10]

High-level intuition. We establish the results by showing that the hardness of LPN implies that of bSVP (for certain parameter settings), which in turn implies EtC-based CRH functions. Informally, assume for contradiction that a useful bSVP solver succeeds in finding a sparse vector $\mathbf{x} = \mathsf{Expand}(\mathbf{y})$ with respect to $n \times q$ matrix $\mathbf{M}$, then this leads to a distinguishing attack against $q = \alpha n$ LPN samples $(\mathbf{M}^\mathsf{T}, \mathbf{M}^\mathsf{T}\mathbf{s} + \mathbf{e})$ by computing

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

which is a biased bit (and thus distinguishable from uniform) due to the sparseness of $\mathbf{x}$. 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]. We now give some informal intuitions on the choices of parameters (omitting constant factors): recall the construction

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

where Expand maps the k-bit strings into q-bit ones of weight up to t. Thus, we essentially need to fulfill the following constraints:

1. $k > n$ (i.e., $h_\mathbf{M}$ is compressing);
2. $k \approx t \log q$ (i.e., $k \approx \log \binom{q}{t}$ for $q \gg t$);
3. The underlying LPN must be at least $2^{\Omega(\mu \cdot t)}$-hard. Recall the distinguishing attack above yield $\mathbf{x}^\mathsf{T}\mathbf{e}$ (the XOR sum of t independent Bernoulli bits of noise μ) that is $(1/2 + 2^{-O(t\mu)})$-biased by the Piling up lemma (Lemma 2) and Stirling's approximation.

Therefore, the parameter choices follow quite naturally for different noise rates. For $\mu = O(1)$, we let sample and time complexity be of the same order, i.e., $\log(q) = \mu \cdot t$, and thus set $t = n^{0.5+\varepsilon}$ and $q = 2^{n^{0.5+\varepsilon}}$, which requires LPN to

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

[10] 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.

be $2^{n^{0.5+\varepsilon}}$-hard for any small $\varepsilon > 0$. However, this does not yield an efficient CRH as the dimensions q and n of $\mathbf{M}$ are not polynomially related, and we solve this problem via a tradeoff between efficiency and security. For $\mu = 1/\sqrt{n}$, we can also use equal sample and time complexity to get parameters $t = 2n^{0.75}$, $q = 2^{n^{0.25}}$ assuming the underlying LPN is $2^{n^{0.25}}$-hard, but this suffers the same efficiency deterioration as in the constant-noise case. Fortunately, with $\mu = 1/\sqrt{n}$ it is possible to avoid this issue by using an alternative choice of parameters: set $q = \mathsf{poly}(n)$ to get polynomially related matrix dimensions and $t = O(n/\log n)$ to satisfy all constraints, which requires LPN of noise rate $1/\sqrt{n}$ to be $2^{\Omega(\sqrt{n}/\log n)}$-hard. The above are the most natural parameter choices for symmetric and asymmetric noise regimes considered in this paper. However, one can use the above methodology to exploit other meaningful parameter settings for any noise rate. For example, for $\mu = O(1)$ set $q = \mathsf{poly}(n)$ and thus requires $t = \Omega(n/\log n)$ and $2^{\Omega(n/\log n)}$-hardness for constant-noise LPN, which is essentially the weak connection established in [5]. Moreover, one can set $q = \mathsf{poly}(n)$, $t = O(n/\log n)$ and assumes polynomial hardness for LPN of extremely low noise rate $\mu = \log^2 n/n$, which corresponds to the noise regime considered in [20].

Practical constructions. Admittedly, the limits of the expand-then-compress methodology (as exposed in the above paragraph) are also obvious: unless under extremely low noise rate [20] the hardness assumed is much beyond polynomial (although still reasonable). Furthermore, for constant noise LPN, the straightforward construction is inefficient and even though standard techniques (by trading security for efficiency) can be applied to make the construction computable in polynomial time, it is still far from practical. We offer an alternative to avoid the efficiency loss, namely, we construct more efficient and polynomially shrinking CRH functions from constant-noise LPN by additionally relying on the idealized random-permutation heuristics (i.e., a block cipher on a public random key behaves like a random permutation). In contrast, most previous blockcipher-based compression functions (e.g. [59,61,15]) 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) [67,56], or require permutations with a large domain (e.g., [31] 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, i.e., $\Pr[\mathcal{B}_\mu = 1] = \mu$, $\Pr[\mathcal{B}_\mu = 0] = 1 - \mu$, 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. We use U_n to denote a uniform distribution over $\{0,1\}^n$ and independent of any other distribution in consideration. $\mathbf{a}\|\mathbf{b}$ 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 $\mathsf{negl}(n) < n^{-N_c}$ for 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.

Definition 1 (Learning Parity with Noise). *The decisional LPN problem with secret length n and noise rate $0 < \mu < 1/2$, denoted by (n,μ)-DLPN, is hard if for every $q = \mathsf{poly}(n)$ and every PPT algorithm $\mathcal{D}$ we have*

$$\left| \Pr[\mathcal{D}(\mathbf{A},\ \mathbf{A}\cdot\mathbf{x}+\mathbf{e}) = 1] - \Pr[\mathcal{D}(\mathbf{A}, U_q) = 1] \right| = \mathsf{negl}(n)\ , \tag{3}$$

and the computational LPN problem with the same n and μ, denoted by (n,μ)-LPN, is hard if for every $q = \mathsf{poly}(n)$ and every PPT algorithm $\mathcal{A}$ we have

$$\Pr[\ \mathcal{A}(\mathbf{A},\ \mathbf{A}\cdot\mathbf{x}+\mathbf{e}) = \mathbf{x}\ ]\ =\ \mathsf{negl}(n)\ , \tag{4}$$

where $q \times n$ matrix $\mathbf{A} \xleftarrow{\$} \{0,1\}^{q\times n}$, $\mathbf{x} \xleftarrow{\$} \{0,1\}^n$ and $\mathbf{e} \leftarrow \mathcal{B}_\mu^q$.

Concrete hardness. *For $T = T(n)$, we say that the (n,μ)-DLPN (resp., (n,μ)-LPN) is T-hard if for every $q \le T$ and every probabilistic adversary of running time T the distinguishing (resp., inverting) advantage in (3) (resp., (4)) is upper bounded by $1/T$. In certain scenario, we use (n,μ, q)-DLPN or (n,μ, q)-LPN, where $q \ll T$ is explicitly stated as a constraint on the number of samples. Finally, note that the T-hardness also applies to other hardness assumptions such as Definition 2 and Definition 3.*

Definition 2 (binary SVP). *For parameters $\alpha = \alpha(n)$ and $\delta = \delta(n)$, the (α, δ)-bSVP assumption asserts that for every probabilistic polynomial time (PPT) algorithm $\mathcal{A}$ it holds that*

$$\Pr_{\mathbf{M}\xleftarrow{\$}\{0,1\}^{n\times\alpha n}}[\ \mathbf{x} \leftarrow \mathcal{A}(\mathbf{M}) \in \{0,1\}^{\alpha n} : 0 < |\mathbf{x}| \le \delta\alpha n \wedge \mathbf{M}\mathbf{x} = \mathbf{0}\ ] = \mathsf{negl}(n)\ .$$

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 [11]. One can just stick to a $h_{\mathbf{M}}$ for some pre-fixed random $\mathbf{M}$.

[11] Recall that a non-uniform attacker can have 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.

Definition 3 (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 $\epsilon > 0$.

The indifferentiability framework [54,23] 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 4 (Indifferentiability [23]). *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 safely replace R in many security scenarios. We refer to the discussions [63,27] on the (in)applicability of indifferentiability results.

Lemma 1 (Chernoff bound). *For any $n \in \mathbb{N}$, let $X_1, \ldots, X_n$ be independent random variables taking values in [0,1] and denote their sum by $\bar{X} = \sum_{i=1}^{n} X_i$. Then, for any $\epsilon > 0$ it holds that*

$$\Pr[\left| \bar{X} - \mathbb{E}[\bar{X}] \right| > n\epsilon] < 2^{-\epsilon^2 \cdot n}\ .$$

Lemma 2 (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 $x \geq 0$ we have $\log(1 + x) \leq x / \ln 2$.*

3 Collision Resistant Hash Functions

3.1 The Expand-then-Compress Construction

We give a high-level overview about the EtC construction from [5] (see Construction 1). Fix a random $n \times \alpha n$ matrix $\mathbf{M}$ which specifies the function. On input $\mathbf{y}$, $h_{\mathbf{M}}$ first stretches (by a factor of b) it into a long-but-sparse vector, i.e., $\mathsf{Expand}(\mathbf{y})$, and then multiply it with $\mathbf{M}$, which compresses by a factor of $1/\alpha$. Thus, the overall shrinkage factor is b/α.

Definition 5. *A function* $\mathsf{Expand} : \{0,1\}^k \to \{0,1\}^\ell$ *is* (b, β)*-expanding if the following conditions are satisfied:*

1. Expand *is injective;*
2. *the expansion factor* $\ell/k \leq b$*;*
3. *and for any* $\mathbf{x} \in \{0,1\}^k$ *the output has Hamming weight* $|\mathsf{Expand}(\mathbf{x})| \leq \beta\ell$*.*

Construction 1 *Let* $k = k(n)$ *and* $\ell = \ell(n)$ *be integer valued functions, and let* $\mathsf{Expand} : \{0,1\}^k \to \{0,1\}^\ell$ *be an expanding function (Definition 5). A collection of functions* $\mathcal{H}_{k,n} = \{h_{\mathbf{M}} : \{0,1\}^k \to \{0,1\}^n, \mathbf{M} \in \{0,1\}^{n\times\ell}\}$ *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* $n \times \ell$ *matrix* $\mathbf{M} \xleftarrow{\$} \{0,1\}^{n\times\ell}$*.*

A couple of proposals on efficient constructions of $\mathsf{Expand}(\cdot)$ were given in [5]. We use the one stated below and reproduce its constructive proof for completeness. The construction assumes WLOG[12] β to be a negative power of 2.

Lemma 3 (The Expand function [5]). *For any integer valued functions* $L = O(\log k)$*,* $\ell = \mathsf{poly}(k)$ *and* $\beta = 2^{-L}$*, there exists an efficient* $(b = \frac{1}{\beta \log(1/\beta)}$*,* $\beta)$*-expanding function* $\mathsf{Expand} : \{0,1\}^k \to \{0,1\}^{\ell(k)}$ *in* AC^0*.*

Proof. For $L = \log(1/\beta)$, Expand parses the k-bit input into L-bit blocks as

$$\mathbf{y} = y_1 \cdots y_L \| y_{L+1} \cdots y_{2L} \| \cdots \| y_{L(k/L-1)+1} \cdots y_k$$

and produces as output

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

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 elsewhere). It is easy to see that the output is of length $\ell = k2^L/L = \frac{k}{\beta \log(1/\beta)}$ and Hamming weight $k/L = \beta\ell$.

[12] For arbitrary β we simply let $\beta' = 2^{-L} \leq \beta < 2^{-L+1}$ and use β' in place of β, which changes the resulting parameters by a factor of less than 2.

Note that the construction is almost optimal as the injection condition of Expand implies $k \leq \log \binom{\ell}{\beta\ell}$, and the construction achieves $k = \beta\ell \log(1/\beta)$, which is very close to the upper bound $\log \binom{\ell}{\beta n} \approx \ell \mathbf{H}(\beta) - \log \ell/2 = \beta\ell(\log(1/\beta) + O(1))$ (see Lemma 4 and Fact 2).

Lemma 4 (Asymptotics for binomial coefficients (e.g. [39], p.492)). *For any $0 < \mu = \mu(n) < 1/2$ we have*

$$\binom{n}{\mu n} = 2^{n\mathbf{H}(\mu) - \frac{\log n}{2} + O(1)}$$

where $\mathbf{H}(\mu) \stackrel{def}{=} \mu \log(1/\mu) + (1-\mu)\log(1/(1-\mu))$ is the binary entropy function.

Fact 2 ([69]) *For any $0 < \mu \leq 1/2$, $\mu \log(1/\mu) < \mathbf{H}(\mu) < \mu(\log(1/\mu) + 3/2)$.*

Theorem 1 (The EtC CRH Construction [5]). *Assume that (α, δ)-bSVP is hard, then Construction 1 instantiated with a (b,β)-expanding function Expand with $b/\alpha < 1$ and $2\beta \leq \delta$ gives rise to a collision-resistant hash function $\mathcal{H}$ with shrinkage factor b/α.*

Proof. Any algorithm that finds out 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 $\mathbf{x} = \mathsf{Expand}(\mathbf{y}) + \mathsf{Expand}(\mathbf{y}') \neq 0$ since the distinctiveness of y and y' are preserved when being applied on any injective function, and

$$|\mathbf{x}| \leq |\mathsf{Expand}(\mathbf{y})| + |\mathsf{Expand}(\mathbf{y}')| \leq 2\beta\alpha n \leq \delta\alpha n \ .$$

On choosing parameters. We explain the intuition on how to choose the parameters for basing CRH functions on bSVP (which in turn relies on LPN). We first substitute the parameters to make the shrinkage factor dependent only on the parameters of bSVP, i.e.,

$$\frac{b}{\alpha} = \frac{1}{\alpha\beta \log(1/\beta)} = \frac{2}{\alpha\delta \log(2/\delta)}$$

where $b = \frac{1}{\beta \log(1/\beta)}$ and $2\beta = \delta$ (see Lemma 3 and Theorem 1). Then, our first construction in Section 3.2 sets $\alpha = n^{2+\epsilon}$ and $\delta = \frac{1}{n^{2+\epsilon}\log n}$ such that the corresponding (α,δ)-bSVP is implied by low-noise LPN, and our second construction in Section 3.3 reduces constant-noise LPN to ($\alpha = \frac{2^{\sqrt{n}}}{n}$, $\delta = \frac{n^{(1+\epsilon)/2}}{2^{\sqrt{n}}}$)-bSVP constant-noise LPN.

3.2 Low-noise LPN Implies CRH functions

We start with an easy and straightforward construction from low-noise LPN.

Theorem 2. *Assume that (n,$\mu = \frac{1}{\sqrt{n}}$, $q = n^{3+\epsilon}$)-DLPN is $2^{4\sqrt{n}/\log n}$-hard for any $\epsilon > 0$, then ($\alpha = n^{2+\epsilon}$, $\delta = \frac{1}{n^{2+\epsilon}\log n}$)-bSVP is $2^{\sqrt{n}/\log n}$-hard and $2^{\sqrt{n}/\log n}$-hard CRH functions with constant shrinkage exist in $\mathsf{AC}^0(\mathsf{MOD2})$.*

Proof. Assume for contradiction there exists an algorithm $\mathcal{A}$ of time $2^{\sqrt{n}/\log n}$ such that the following holds for infinitely many n's:

$$\Pr[\ \mathbf{x} \leftarrow \mathcal{A}(\mathbf{M}) \in \{0,1\}^{n^{3+\epsilon}} : 0 < |\mathbf{x}| \le \frac{n}{\log n} \wedge \mathbf{M}\mathbf{x} = \mathbf{0}\ ] \ge 2^{-\frac{\sqrt{n}}{\log n}} \ ,$$
$$\mathbf{M} \xleftarrow{\$} \{0,1\}^{n \times n^{3+\epsilon}}$$

then $\mathcal{A}$ can be used to solve the $(n, \mu = \frac{1}{\sqrt{n}}, q = n^{3+\epsilon})$-DLPN problem. That is, apply distinguisher $\mathcal{D}_1$ (as defined in Algorithm 1) on input $(\mathbf{M}^\mathsf{T}, \mathbf{z})$, where $\mathbf{M}^\mathsf{T} \xleftarrow{\$} \{0,1\}^{n^{3+\epsilon} \times n}$, and either $\mathbf{z} = \mathbf{M}^\mathsf{T}\mathbf{s} + \mathbf{e}$ or $\mathbf{z} \xleftarrow{\$} \{0,1\}^{n^{3+\epsilon}}$, $\mathbf{s} \xleftarrow{\$} \{0,1\}^n$ and $\mathbf{e} \leftarrow \mathcal{B}_\mu^{n^{3+\epsilon}}$. When $\mathbf{z} = \mathbf{M}^\mathsf{T}\mathbf{s} + \mathbf{e}$ and $\mathcal{A}$ succeeds in finding such $\mathbf{x}$, we have by Lemma 2 and Fact 1

$$\Pr[\mathbf{x}^\mathsf{T}\mathbf{z} = \mathbf{x}^\mathsf{T}\mathbf{e} = 0] = \frac{1}{2} + 2^{\frac{-c\mu n}{\log n} - 1} \ge \frac{1}{2} + 2^{-\frac{\frac{2}{\sqrt{n}}}{\ln 2(1-\frac{2}{\sqrt{n}})} \frac{n}{\log n} - 1} \ge \frac{1}{2} + 2^{-\frac{3\sqrt{n}}{\log n}} \ .$$

Therefore, $\mathcal{D}_1$ achieves advantage

$$\begin{aligned} &\Pr[\mathcal{D}_1(\mathbf{M}^\mathsf{T}, \mathbf{M}^\mathsf{T}\mathbf{s} + \mathbf{e}) = 0] - \Pr[\mathcal{D}_1(\mathbf{M}^\mathsf{T}, U_{n^{3+\epsilon}}) = 0] \\ &\ge 2^{-\frac{\sqrt{n}}{\log n}} \cdot 2^{-\frac{3\sqrt{n}}{\log n}} \ge 2^{-\frac{4\sqrt{n}}{\log n}} \ , \end{aligned}$$

which is a contradiction to the assumption. We have by Theorem 1 and Lemma 3 that there exists a $2^{\sqrt{n}/\log n}$-hard CRH function $\mathcal{H} = \{h_\mathbf{M} : \{0,1\}^k \to \{0,1\}^n\}$ with $2\beta = \delta = \frac{1}{n^{2+\epsilon}\log n}$, $b = \frac{1}{\beta \log(1/\beta)}$, $\alpha = n^{2+\epsilon}$ and shrinkage factor

$$\frac{n}{k} \le \frac{b}{\alpha} = \frac{1}{\alpha\beta\log(1/\beta)} = \frac{2n^{2+\epsilon}\log n}{n^{2+\epsilon}\big((2+\epsilon)\log n + \log\log n + 1\big)} < \frac{2}{2+\epsilon} \ .$$

Algorithm 1 A distinguisher $\mathcal{D}_1$ for $(n, \mu = \frac{1}{\sqrt{n}}, q = n^{3+\epsilon})$-DLPN

Input: $(\mathbf{M}^\mathsf{T}, \mathbf{z})$, where $\mathbf{M}^\mathsf{T} \in \{0,1\}^{n^{3+\epsilon} \times n}$ and $\mathbf{z} \in \{0,1\}^{n^{3+\epsilon}}$
$\mathbf{x} \leftarrow \mathcal{A}(\mathbf{M})$;
if $0 < |\mathbf{x}| \le \frac{n}{\log n} \wedge \mathbf{M}\mathbf{x} = \mathbf{0}$ **then**
  $v = \mathbf{x}^\mathsf{T}\mathbf{z}$
else
  $v \xleftarrow{\$} \{0,1\}$
end if
Output: v

3.3 Constant-noise LPN Implies CRH functions

The case of constant-noise LPN is slightly more complicated. The reduction from LPN to bSVP (see Lemma 6) is mostly adapted from Theorem 2, but a direct construction of CRH from bSVP seems hard, and we (see Theorem 3) establish the feasibility via a tradeoff between efficiency and security.

Theorem 3. *Assume that (n,μ)-DLPN is $2^{n^{1/2+\epsilon}}$-hard for any constants $0 < \mu < 1/2$ and $\epsilon > 0$, then CRH functions with poly-logarithmic shrinkage factors exist in* $\mathsf{AC}^0(\mathsf{MOD2})$.

Proof. It follows from Lemma 6 that $2^{n^{1/2+\epsilon}}$-hard constant-noise DLPN implies $2^{\frac{n^{1/2+\epsilon}}{3}}$-hard ($\alpha = \frac{2^{\sqrt{n}}}{n}$, $\delta = \frac{n^{(1+\epsilon)/2}}{2^{\sqrt{n}}}$)-bSVP, which in turns (by Theorem 1 and Lemma 3) implies for $\alpha = \frac{2^{\sqrt{n}}}{n}$ and $2\beta = \delta = \frac{n^{(1+\epsilon)/2}}{2^{\sqrt{n}}}$ there exists $2^{\frac{n^{1/2+\epsilon}}{3}}$-hard $\mathcal{H} = \{h_{\mathbf{M}} : \{0,1\}^k \to \{0,1\}^n, \mathbf{M} \in \{0,1\}^{n \times 2^{\sqrt{n}}}\}$ with

$$\frac{n}{k} \leq \frac{b}{\alpha} = \frac{1}{\alpha\beta \log(1/\beta)} = \frac{1}{\Omega(n^{\epsilon/2})} \ .$$

However, we have to use $\lambda = 2^{\sqrt{n}}$ (instead of n as $h_{\mathbf{M}}$ is not computable in $\mathsf{poly}(n)$-time) as the main security parameter and represent other parameters as functions of λ, e.g., $n = \log^2 \lambda$ and $k = \Omega(n^{1+\epsilon/2}) = \Omega(\log^{2+\epsilon} \lambda)$. That is, $\mathcal{H} = \{h_{\mathbf{M}} : \{0,1\}^{\Omega(\log^{2+\epsilon} \lambda)} \to \{0,1\}^{\log^2 \lambda}, \mathbf{M} \in \{0,1\}^{\log^2 \lambda \times \lambda}\}$ is a $\lambda^{\frac{\log^{2\epsilon} \lambda}{3}}$-hard CRH function, which (by Lemma 5) implies a domain/range-extended CRH $\mathcal{H}' = \{h'_{\mathbf{M}} : \{0,1\}^{\Omega(\lambda \log^{\epsilon} \lambda)} \to \{0,1\}^{\lambda}, \mathbf{M} \in \{0,1\}^{\log^2 \lambda \times \lambda}\}$.

Lemma 5 (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.

In the proof of Lemma 6 below, we show a stronger reduction that any algorithm that breaks the bSVP implies another algorithm that breaks the decisional LPN problem with time complexity $2^{n^{1/2+\epsilon}}$ and advantage nearly 1/2 (instead of $2^{-n^{1/2+\epsilon}}$ as needed by the statement).

Lemma 6. *Assume that (n,μ)-DLPN is $2^{n^{1/2+\epsilon}}$-hard for any constants $0 < \mu < 1/2$ and $\epsilon > 0$, then ($\alpha = \frac{2^{\sqrt{n}}}{n}$, $\delta = \frac{n^{(1+\epsilon)/2}}{2^{\sqrt{n}}}$)-bSVP is $2^{\frac{n^{1/2+\epsilon}}{3}}$-hard.*

Proof. Assume for contradiction there exists an algorithm $\mathcal{A}$ of time $2^{n^{1/2+\epsilon}}/3$ s.t. the following holds for infinitely many n's:

$$\Pr_{\mathbf{M} \xleftarrow{\$} \{0,1\}^{n \times 2^{\sqrt{n}}}}[\ \mathbf{x} \leftarrow \mathcal{A}(\mathbf{M}) \in \{0,1\}^{2^{\sqrt{n}}} : 0 < |\mathbf{x}| \leq n^{1/2+\epsilon/2} \wedge \mathbf{M}\mathbf{x} = \mathbf{0} \ ] \geq 2^{-\frac{n^{1/2+\epsilon}}{3}} \ ,$$

then we show that it implies a distinguisher $\mathcal{D}_2$ (see Algorithm 2) that solves (with constant advantage) the constant-noise DLPN problem (for any $0 < \mu < 1/2$) using sample complexity $q < 2^{n^{1/2+\epsilon}}$ and time complexity $T < 2^{n^{1/2+\epsilon}}$. As

stated in Algorithm 2, if $\mathcal{D}_2$ is applied to $(\mathbf{A}, \mathbf{z} = \mathbf{As}+\mathbf{e})$, where $\mathbf{A} \xleftarrow{\$} \{0,1\}^{q\times n}$ and $\mathbf{e} \leftarrow \mathcal{B}_\mu^q$, then every i-th vote v_i satisfies (for all large enough n's)

$$\begin{aligned}\Pr[v_i = 0] &= \Pr[\mathcal{E}_i]\Pr[\mathbf{x}^\mathsf{T}\mathbf{z} = 0] + \frac{\Pr[\neg\mathcal{E}_i]}{2} \\ &\geq \frac{1}{2} + 2^{\frac{-n^{\frac{1}{2}+\epsilon}}{3}} \cdot 2^{-c_\mu n^{\frac{1+\epsilon}{2}}-1} > \frac{1}{2} + 2^{-0.4n^{\frac{1}{2}+\epsilon}} \quad .\end{aligned}$$

where $\mathcal{E}_i$ denotes the event that $\mathcal{A}$ succeeds on $\mathbf{M}_i$ and c_μ is a constant dependent on μ (see Lemma 2). It follows by the Chernoff bound (Lemma 1) that

$$\Pr[\mathcal{D}_2(\mathbf{A}, \mathbf{As}+\mathbf{e}) = 0] \geq 1 - 2^{-n}$$

and in contrast $\Pr[\mathcal{D}_2(\mathbf{A}, U_q) = 0] = 1/2$, which completes the proof.

Algorithm 2 A distinguisher $\mathcal{D}_2$ for constant-noise DLPN

Input: $(\mathbf{A}, \mathbf{z})$, where $\mathbf{A} \in \{0,1\}^{q\times n}$, $\mathbf{z} \in \{0,1\}^q$, $l = n2^{0.8n^{\frac{1}{2}+\epsilon}}$ and $q = l \cdot 2^{\sqrt{n}}$
parse $\mathbf{A}^\mathsf{T}$ as a number of $n \times 2^{\sqrt{n}}$ matrices $\mathbf{M}_1, \mathbf{M}_2, \cdots, \mathbf{M}_l$
parse $\mathbf{z}$ as $2^{\sqrt{n}}$-bit blocks $\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_l$ accordingly
for $i = 1$ to l **do**
  $\mathbf{x}_i \leftarrow \mathcal{A}(\mathbf{M}_i)$;
  if $0 < |\mathbf{x}_i| \leq n^{1/2+\epsilon/2} \wedge \mathbf{M}_i\mathbf{x}_i = \mathbf{0}$ **then**
    $v_i = \mathbf{x}_i^\mathsf{T}\mathbf{z}_i$
  else
    $v_i \xleftarrow{\$} \{0,1\}$
  end if
end for
Output: the majority bit of $v_1, \cdots, v_l$

3.4 More Efficient Heuristic-based CRH from Constant-Noise LPN

Notice that the CRH immediately implied by constant-noise LPN (see Lemma 6) 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 (in the security parameter n). This motivates us (see the proof of Theorem 3) to switch to another parameter $\lambda = 2^{\sqrt{n}}$ such that hash function is 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, Theorem 3 mainly establishes feasibility results about basing CRH on constant-noise LPN. We discuss an alternative to void the loss, i.e., to preserve security, polynomial shrinkage and efficiency at the same time. This relies on idealized assumptions (i.e., a block cipher keyed with a random public string behaves like a random permutation) in addition to constant-noise LPN.

THE INTUITION. We recall that the CRH function $h_{\mathbf{M}}(\mathbf{y}) = \mathbf{M} \cdot \mathsf{Expand}(\mathbf{y})$ for an $n \times \alpha n$ matrix $\mathbf{M}$ and that Expand parses $\mathbf{y}$ into k/L blocks and produces same number of output blocks accordingly, where $\alpha n = k2^L/L$. We also parse $\mathbf{M}$ into k/L equal-size submatrices $\mathbf{M}_1, \cdots, \mathbf{M}_{k/L}$, each of dimension $n \times 2^L$. Let $R : \{0,1\}^{\log(\alpha n)} \to \{0,1\}^n$ be a random function that describes $\mathbf{M}$, i.e., for every $j \in \{0,1\}^{\log(\alpha n)}$ the output $R(j)$ corresponds to the j-th column of $\mathbf{M}$. We thus have

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

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 $R : \{0,1\}^{\log(\alpha n)} \to \{0,1\}^n$ (recall $\log(\alpha n) \ll n$). One may want to replace R with a pseudorandom function (with key made public), but in general the security cannot be argued in the standard model due to the distinction between public-coin and secret-coin CRH functions [43].

In order to explore more efficient constructions, we use more aggressive yet still reasonable (in respect of the current state-of-the-art attacks [17]) hardness assumption about LPN to obtain Lemma 7 below.

Lemma 7. *Assume that (n,μ)-DLPN is $2^{\frac{5}{4}n^{0.8}}$-hard under noise rate $\mu = 1/4$, then $(\alpha = \frac{2^{n^{0.8}}}{n}, \delta = \frac{n^{0.8}}{2^{n^{0.8}+2}})$-bSVP is $2^{n^{0.8}}$-hard. Further, assume in addition that $R : \{0,1\}^{n^{0.8}} \to \{0,1\}^n$ behaves like a random function, then $h_R : \{0,1\}^{\frac{n^{1.6}}{9}} \to \{0,1\}^n$ defined as below is a $2^{n^{0.8}}$-hard CRH function, where*

$$h_R(\mathbf{y}) = \bigoplus_{i=1}^{\frac{n^{0.8}}{9}} R(i\|\mathbf{y}_i) \ ,$$

and input $\mathbf{y}$ is parsed as $n^{0.8}$-bit blocks $\mathbf{y}_1, \cdots, \mathbf{y}_{n^{0.8}/9}$.

Proof. The proof of the first statement is almost the same (up to choices of parameters) as Theorem 2 and Lemma 6 and we thus omit the redundancy. The statement about CRH follows from Theorem 1 and (5), where parameters k for $h_R : \{0,1\}^k \to \{0,1\}^n$ and L as in Lemma 3 satisfy the following (for all $n \geq 128$):

$$\frac{n}{k} \leq \frac{2}{\alpha\delta\log(2/\delta)} = \frac{8n^{0.2}}{n^{0.8}+3-0.8\log n} \leq \frac{9}{n^{0.6}} \ ,$$

$$n^{0.8} \geq \Big(L = \log(2/\delta) = n^{0.8}+3-0.8\log n\Big) \geq 0.9n^{0.8} \ ,$$

which completes the proof.

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 $\pi \oplus \pi^{-1}$ [30] (or $\pi_1 \oplus \pi_2$ [57]) 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 $l = n - \log(\alpha n)$ 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 state below the indifferentiability of $\pi \oplus \pi^{-1}$ from a random function by Dodis et al. [30].

Lemma 8 (Lemma 4 from [30]). *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.*

SECURITY AND EFFICIENCY. With Lemma 7 and Lemma 8 now we can have a rough estimation about the (asymptotic) security of the blockcipher-based CRH. Assume that the LPN problem for noise rate $\mu = 1/4$ is $2^{\frac{5}{4}n^{0.8}}$-hard and a block cipher (with a random key) is a perfect instantiation of a random permutation, then any adversary whose running time (and also q) is bounded by, say $2^{0.8n^{0.8}}$, succeeds in finding a collision for the CRH function with probability less than

$$2^{-0.8n^{0.8}} + O(1) \cdot 2^{1.6n^{0.8}-n} \quad .$$

Moreover, the CRH function compresses $\frac{n^{1.6}}{9}$ bits into n bits with a shrinkage fact $\frac{9}{n^{0.6}}$ and invokes the underlying block cipher $2n^{0.8}/9$ times.

ASYMPTOTIC VS. CONCRETE. The above bounds are asymptotic, whose concrete values are meaningful only for "all sufficiently large n". For modern block cipher instantiations, $n = 128$ seems not large enough as: 1) we are not sure if LPN on secret size $n = 128$ and noise rate $1/4$ can have $\frac{5}{4}n^{0.8} \approx 60$ bits of security; 2) for $n = 128$ the shrinkage factor $\frac{9}{n^{0.6}} \approx 1/2$ does not reach the large rate as can be expected from "polynomial shrinkage". Fortunately, there are a few cryptographic permutation candidates of larger sizes (e.g., $n = 400, 800, 1600$), such as the permutation family underlying SHA-3 [41] and the Simpira family [40], for which our constructions can offer efficient and parallel domain extensions (by a polynomial factor). Further, just like most provable security statements, our results provide only the lower bound of security that "the CRH is at least as hard as the underlying LPN", which could be quite loose. A similar situation is the

case of SWIFFT [53], where the authors of [53] showed that SWIFFT is at least as hard as an ideal lattice problem, but SWIFFT turns out to be significantly harder than its underlying instantiation of the lattice problem (which can be broken in a moderate amount of time).

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. [59,61,15], 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) [67,56], or require permutations with a large domain (e.g., [31] 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 resolve the open problem whether CRH functions can be based on the hardness of LPN and we show that under commonly believed hardness assumptions about LPN efficient CRH functions exist in polynomial-size constant-depth circuits. 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.: eBASH: Ecrypt benchmarking of all submitted hashes (2011), http://bench.cr.yp.to
12. Bernstein, D.J., Lange, T., Niederhagen, R., Peters, C., Schwabe, P.: FSBday:implementing wagner's generalized birthday attack against the sha3 round1 candidate fsb. In: Progress in Cryptology - INDOCRYPT 2009. pp. 18–38 (2009)
13. Bernstein, D.J., Lange, T., Peters, C.: Smaller decoding exponents: Ball-collision decoding. In: Advances in Cryptology - CRYPTO 2011. pp. 743–760 (2011)
14. Bernstein, D.J., Lange, T., Peters, C., Schwabe, P.: Really fast syndrome-based hashing. In: Progress in Cryptology - AFRICACRYPT 2011. pp. 134–152 (2011)
15. 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)
16. 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)
17. 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)
18. Brakerski, Z.: When homomorphism becomes a liability. In: Proceedings of 10th Theory of Cryptography Conference (TCC 2013). pp. 143–161 (2013)
19. 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
20. 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)
21. 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)
22. 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)
23. Coron, J.S., Dodis, Y., Malinaud, C., Puniya, P.: Merkle-damgård revisited: How to construct a hash function. In: Shoup [66], pp. 430–448
24. Damgård, I.: Collision free hash functions and public key signature schemes. In: Advances in Cryptology - EUROCRYPT '87. pp. 203–216 (1987)
25. Damgård, I.: A design principle for hash functions. In: Advances in Cryptology - CRYPTO '89. pp. 416–427 (1989)
26. 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)

27. 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)
28. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Transactions on Information Theory IT-22(6), 644–654 (1976)
29. 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)
30. 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)
31. 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)
32. Döttling, N.: Low noise lpn: Kdm secure public key encryption and sample amplification. In: Public-Key Cryptography–PKC 2015. pp. 604–626. Springer (2015)
33. 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)
34. 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)
35. 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)
36. Finiasz, M., Gaborit, P., Sendrier, N.: Improved fast syndrome based cryptographic hash functions. In: Proceedings of ECRYPT hash workshop 2007 (2007), http://finiasz.net/research/2007/finiasz-gaborit-sendrier-ecrypt-hash-workshop07.pdf
37. Fouque, P., Leurent, G.: Cryptanalysis of a hash function based on quasi-cyclic codes. In: Topics in Cryptology - CT-RSA 2008. pp. 19–35 (2008)
38. 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)
39. Graham, R.L., Knuth, D.E., Patashnik, O.: Concrete Mathematics: A Foundation for Computer Science. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edn. (1994)
40. Gueron, S., Mouha, N.: Simpira v2: A Family of Efficient Permutations Using the AES Round Function. In: Cheon, J.H., Takagi, T. (eds.) Advances in Cryptology – ASIACRYPT 2016, Part I, Lecture Notes in Computer Science, vol. 10031, pp. 95–125. Springer Berlin Heidelberg (2016)
41. Guido Bertoni, Joan Daemen, M.P., Assche, G.V.: The keccak reference. Submission to NIST (Round 3) (2011), http://keccak.noekeon.org/Keccak-reference-3.0.pdf.
42. 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)

43. 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)
44. Impagliazzo, R.: A personal view of average-case complexity. In: Structure in Complexity Theory Conference. pp. 134–147 (1995)
45. 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)
46. Juels, A., Weis, S.A.: Authenticating pervasive devices with human protocols. In: Shoup [66], pp. 293–308
47. 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)
48. 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)
49. 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)
50. Kirchner, P.: Improved generalized birthday attack. Cryptology ePrint Archive, Report 2011/377 (2011), http://eprint.iacr.org/2011/377
51. 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)
52. 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)
53. 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)
54. 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)
55. 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)
56. 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)
57. 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)
58. Merkle, R.: Secrecy, Authentication, and Public Key Systems. Ph.D. thesis (1979)

59. Merkle, R.C.: One way hash functions and DES. In: Advances in Cryptology - CRYPTO '89. pp. 428–446 (1989)
60. Pietrzak, K.: Cryptography from learning parity with noise. In: Proceedings of the Theory and Practice of Computer Science (SOFTSEM 2012). pp. 99–114 (2012)
61. 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)
62. 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)
63. 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)
64. 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)
65. Saarinen, M.O.: Linearization attacks against syndrome based hashes. In: Progress in Cryptology - INDOCRYPT 2007. pp. 1–9 (2007)
66. Shoup, V. (ed.): Advances in Cryptology—CRYPTO 2005, LNCS, vol. 3621. Springer-Verlag (14–18 Aug 2005)
67. 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)
68. Vardy, A.: The intractability of computing the minimum distance of a code. IEEE Transactions on Information Theory 43(6), 1757–1766 (1997)
69. Yu, Y., Zhang, J.: Cryptography with auxiliary input and trapdoor from constant-noise lpn. In: Advances in Cryptology – CRYPTO 2016. pp. 214–243 (2016)