Low-Complexity Cryptographic Hash Functions*

Benny Applebaum
Tel-Aviv University

Naama Haramaty
Technion

Yuval Ishai
Technion and UCLA

Eyal Kushilevitz
Technion

Vinod Vaikuntanathan
MIT

Abstract

Cryptographic hash functions are efficiently computable functions that shrink a long input into a shorter output while achieving some of the useful security properties of a random function. The most common type of such hash functions is *collision resistant* hash functions (CRH), which prevent an efficient attacker from finding a pair of inputs on which the function has the same output.

Despite the ubiquitous role of hash functions in cryptography, several of the most basic questions regarding their computational and algebraic complexity remained open. In this work we settle most of these questions under new, but arguably quite conservative, cryptographic assumptions, whose study may be of independent interest. Concretely, we obtain the following results:

- **Low-complexity CRH.** Assuming the intractability of finding short codewords in natural families of linear error-correcting codes, there are CRH that shrink the input by a constant factor and have a *constant algebraic degree* over $\mathbb{Z}_2$ (as low as 3), or even *constant output locality and input locality*. Alternatively, CRH with an arbitrary polynomial shrinkage can be computed by *linear-size* circuits.
- **Win-win results.** If low-degree CRH with good shrinkage *do not* exist, this has useful consequences for learning algorithms and data structures.
- **Degree-2 hash functions.** Assuming the conjectured intractability of solving a random system of quadratic equations over $\mathbb{Z}_2$, a uniformly random degree-2 mapping is a *universal one-way hash function* (UOWHF). UOWHF relaxes CRH by forcing the attacker to find a collision with a random input picked by a challenger. On the other hand, a uniformly random degree-2 mapping is *not* a CRH. We leave the existence of degree-2 CRH open, and relate it to open questions on the existence of degree-2 randomized encodings of functions.

*A preliminary version of this paper has appeared in the 8th Innovations in Theoretical Computer Science (ITCS), 2017.

1 Introduction

This work studies the problem of minimizing the complexity of cryptographic hash functions. We start with some relevant background.

Cryptographic hash functions are efficiently computable functions that shrink a long input into a shorter output while achieving some of the useful security properties of a random function. The main focus of this work is on *collision resistant* hash functions (CRH), which prevent an efficient attacker from finding a pair of distinct inputs x, x' on which the function has the same output.[1] However, we will also consider *universal one-way hash function* (UOWHF) [63], which relax CRH by forcing the attacker to find a collision with a random input x picked by a challenger.

CRH are among the most useful and well studied cryptographic primitives. They are commonly used in cryptographic protocols, with applications ranging from sublinear-communication and statistically hiding commitments [29, 47], via succinct and efficiently verifiable arguments for NP [54, 62], to protocols that bypass black-box simulation barriers [7]. More directly, they can be used via the "hash and sign" paradigm to reduce the task of digitally signing a long message x to the easier task of signing a short hash $h(x)$ [28, 61]. Analogously, they can reduce the cost of verifying the correctness of a long NP-statement x to that of verifying the correctness of a short NP-statement $y = h(x)$ by having the prover argue that she knows some x' such that $h(x') = y$ and x' is a true statement. Thus, the amortized cost of signing a long message or verifying a long NP-statement is essentially the cost of computing a CRH.

While the *feasibility* of CRH can be based on a variety of standard cryptographic assumptions, including the conjectured intractability of factoring, discrete logarithms, and lattice problems [28, 42, 65, 58], questions about the *efficiency* of CRH are still quite far from being settled. In particular, recent progress on the efficiency of other "symmetric" cryptographic primitives, such as pseudorandom generators, [20, 75], pseudorandom functions [43], and even UOWHFs, does not seem relevant in light of the known black-box separation between CRH and these primitives [69, 45]. The goal of the present work is to close some of the remaining gaps in our understanding of the complexity of CRH and related primitives.

We study the following natural complexity measures:

- **Degree.** We say that $h : \{0,1\}^k \to \{0,1\}^m$ has algebraic degree d if each output can be written as a multivariate polynomial over $\mathbb{Z}_2$ in the inputs of degree at most d. Ideally, we would like the degree to be constant, where 2 is the best one could hope for.

- **Locality.** We say that h has *output locality* d if each output depends on at most d inputs. Ideally, we would like the output locality to be constant, where output locality 3 is the best one could hope for [41]. If h has output locality d then its degree is at most d. Similarly, h has *input locality* d if every input influences at most d outputs.

- **Circuit size.** We say that h has circuit size S if it can be computed by a boolean circuit of size S over the standard AND/OR/NOT basis (with AND/OR gates of fan-in 2).[2] Ideally, we would like the circuit size to be *linear* in the input length. Linear size is implied by constant output locality.

[1] Technically speaking, a CRH is defined by a collection of input-shrinking functions h_z, where z is a public evaluation key, and where the security requirement should hold with respect to a randomly chosen z. This has the advantage of allowing security against non-uniform attackers. In the following presentation we will treat a CRH as a single deterministic function for simplicity.

[2] One could alternatively consider *running time* on a RAM machine; our upper bounds on circuit size apply to this model as well.

The goals of minimizing circuit size and locality can be directly motivated by the goals of reducing the sequential and parallel time complexity of hashing. Minimizing algebraic degree, other than being of theoretical interest, is motivated by applications in which hashing is computed in the encrypted or secret-shared domain. Indeed, it is typically the case that techniques for secure multiparty computation [12, 26, 66], homomorphic encryption [39, 22, 40], or homomorphic secret sharing [27, 21] are much more efficient when applied to low-degree computations over a small field. See [46] for further discussion.

The prior state of the art can be summarized as follows. Standard algebraic or number theoretic constructions of CRH, as well as (asymptotic versions of) the commonly used practical designs, do not achieve constant degree or locality, and their circuit size is quasi-linear or worse. General techniques for randomized encoding of functions can be used to convert any standard CRH h in NC^1 into a CRH $\hat{h}$ with constant output locality [1]. However, even if h has very good shrinkage, $\hat{h}$ only shrinks the input by a sublinear amount, which limits its usefulness. From here on, we will restrict the attention by default to hash functions that have linear (or better) shrinkage, namely $h : \{0,1\}^k \to \{0,1\}^{ck}$ for some $0 < c < 1$. Every such h with linear circuit size can be converted into a linear-size CRH with polynomial shrinkage, namely $h' : \{0,1\}^k \to \{0,1\}^{k^\epsilon}$ for an arbitrary $\epsilon > 0$, using a tree of invocations of h [60]. Linear-size UOWHFs were constructed in [52] under strong assumptions. The assumptions were later improved in [3], who also achieved constant locality. The question of obtaining similar results for CRH was left open by both works. Finally, heuristic constructions of CRH with constant algebraic degree have been proposed in the literature [30]. However, the security of these proposals has not been reduced to a well studied problem.

To summarize, prior to our work, linear-shrinkage CRH candidates with constant algebraic degree were only proposed as heuristics, and no candidate CRH construction with constant locality or linear circuit size has been proposed.

1.1 Our Contribution

In this work we settle most of the open questions concerning the complexity of CRH and related primitives under new, but arguably clean and conservative, cryptographic assumptions.

Concretely, we put forward the following class of *binary SVP* assumptions. For a distribution $\mathcal{M}$ over $m \times n$ binary matrices and a parameter $0 < \delta < 1/2$, the $(\mathcal{M}, \delta) - \mathsf{bSVP}$ assumption asserts that given a matrix M drawn from $\mathcal{M}$, no efficient algorithm can find a nonzero vector in the kernel of M whose Hamming weight is at most δn, except with negligible success probability. The matrix M can be thought of as the parity-check matrix of a binary linear error-correcting code. Thus, bSVP can be viewed as a binary field analogue of the lattice Shortest Vector Problem (SVP), replacing an integer lattice by a binary code.

We construct low-complexity CRH based on instances of the bSVP assumption with matrix distributions $\mathcal{M}$ that correspond to uniform distributions over natural classes of linear codes. The parameter δ is chosen such that there are exponentially many codewords whose relative weight is close to δ, but where such codewords are only an exponentially small fraction of the set of all codewords, thus ruling out "guessing attacks." When $m = \alpha n$ and $\mathcal{M}$ is sufficiently rich (in particular, when it is uniform over *all* $m \times n$ matrices), the assumption is plausible whenever $\delta < \alpha/2$. The assumption does *not* hold when $\delta > \alpha/2$, since in this case a codeword of weight δn can be found by solving a system of linear equations.

Despite being a simple and natural cryptographic assumption, we are not aware of any explicit study or even precise formulation of the bSVP assumption in the literature. While we were not able to reduce useful instances of bSVP to any standard cryptographic assumption, we do show that such instances have a "win-win" flavor in the sense that if they are broken, this would necessarily

have useful algorithmic consequences. Natural instances of the bSVP assumption are likely to find additional applications in cryptography, and their further study may be of independent interest from both a cryptography and coding theory points of view.

We now give a more detailed account of our results.

***Low-complexity CRH.** Assuming bSVP for a random linear code with $\delta > 2H_2^{-1}(\alpha)$ (where H_2 denotes the binary entropy function), there are CRH that shrink the input by a constant factor and have a *constant algebraic degree*. We give a direct construction of degree-5 CRH, and then reduce the degree to 3 by using a new optimized randomized encoding construction for constant-degree functions (previous randomized encoding methods from [1] can also reduce the degree to 3, but at the expense of compromising the linear shrinkage feature).

Assuming bSVP for a random low-density parity-check code (LDPC), we can also get *constant output and input locality*, which imply CRH with an arbitrary polynomial shrinkage that can be computed by *linear-size* circuits. The assumption that bSVP holds for LDPCs may look too strong in light of the fact that LDPCs admit efficient decoding algorithms. However, known decoding techniques seem to have only limited relevance to bSVP. Indeed, the known reductions from bSVP to unique decoding introduce exponential overhead (cf. [32]). Moreover, there is a gap between the noise level p for which LDPC's admit efficient decoding and the relative distance Δ of LDPC's which essentially corresponds to our parameter δ. This gap grows with the (constant) locality parameter [23], and the LDPC becomes similar to random linear code both combinatorially [37, 57], and, presumably, in terms of its intractability.[3]

Our constructions take the following natural high level approach. First the input is deterministically encoded into a longer vector that has a low weight. This encoding is done via a simple function Expand that has constant input and output locality. Then the encoded input is shrunk by applying a random linear mapping M sampled from $\mathcal{M}$, where M is used as a key specifying the CRH. Finding a collision implies finding a low-weight vector in the kernel of M (namely, the sum of two images of Expand), which is intractable if the appropriate instance of the bSVP assumption holds. A practically-oriented hash function candidate with a similar structure was proposed by Augot et al. [4] (see also [35]). In fact, our degree-5 construction can be obtained as an instance of their construction (with a specific choice of parameters). Our other instantiations of this approach are different and are tailored to different optimization goals.

As an application, our linear-size CRH imply (together with other cryptographic assumptions, cf. [16]) the first succinct non-interactive argument system for NP in which the verifier's algorithm can be implemented by a linear-size circuit in the statement length. They also imply the first linear-size implementations of non-interactive *statistically hiding commitments* (SHC), a randomized variant of CRH that can be used to hide the input. This follows from the known constructions of SHC from CRH [29, 47].

***Win-win results.** To gain more insight on the instances of the bSVP assumption on which we rely, we show that refuting them would have useful algorithmic consequences. Concretely, we show two types of such results. First, we show that either (1) there is a linearly-shrinking CRH with logarithmic degree (a non-trivial object that does not seem to follow from standard assumptions) or (2) one can achieve an arbitrary polynomial speedup over the celebrated BKW algorithm for Learning Parities with Noise (LPN) [19]. The latter would be considered a breakthrough in light

[3] We further mention that the problem of finding (many) w-weight codewords in LDPC with sub-constant rate (e.g., when the parity check matrix has m rows and $n = O(m^{7/5})$ columns and $w = O(m^{0.2})$) was implicitly considered by Feige, Kim and Ofek [34]. In particular, it was shown that if the problem is easy (for randomly chosen 3-sparse ,matrices) then one can efficiently *refute* random 3-CNF's with m variables and $m^{1.4}$, beating the state-of-the-art refutation algorithms.

of the large body of work on algorithms for LPN and its variants. Second, we show that breaking useful instances of bSVP, on which a degree-3 linearly-shrinking CRH can be based, leads to a surprisingly good *data structure* for learning parities from random (noiseless) examples in a natural distributed learning model.

***Degree-2 hash functions.** Finally, we study the case of hash functions that have the minimal possible degree. We first address the case of UOWHFs, showing that a random shrinking degree-2 mapping is a UOWHF assuming that it is one-way. The latter is equivalent to a fairly well studied assumption, known as the "MQ assumption" [59, 73], which asserts that solving a random system of quadratic equations is intractable.

We then show several results on the existence of a degree-2 CRH. We show that a *random* degree-2 shrinking function is not collision resistant, strengthening a claim from [30] that was restricted to the case of linear-shrinkage. This result can be extended to the case of SHC, leaving open the possibility of constructing degree-2 CRH and SHC by using other distributions over degree-2 mappings.

We relate this question to questions on the existence of degree-2 randomized encodings of functions that were left open by [51]. The high level idea is that while for strong version of randomized encoding the existence of degree-2 encodings for general functions can be ruled out, there are relaxed versions for which this question is still open, yet these relaxed versions are strong enough to respect the security properties of CRH and SHC. Thus, ruling out a degree-2 implementation of these primitives would require settling the above open questions in the negative.

*Organization. Following some preliminaries (Section 2), in Section 3 we discuss the assumptions on which we rely, including the bSVP assumption we introduce and the MQ assumption. In Section 4 we present constructions of low-complexity CRH from variants of bSVP. In Section 5 we present our positive and negative results for degree-2 hash functions. Finally, in Section 6 we present the "win-win" results showing that if low-complexity CRH do not exist, this has useful algorithmic consequences.

2 Preliminaries

*General. We let $[n]$ denote the set $\{1, \ldots, n\}$. We naturally view n-bit strings as (column) vectors over the binary field $\mathbb{Z}_2$. For a pair of strings $x, x' \in \{0,1\}^n$, we let $\Delta(x, x')$ denote the relative Hamming distance between x and x', i.e., $|\{i \in [n] : x_i \neq x'_i\}|/n$. We let $\Delta(x)$ denote the (relative) Hamming weight of x, i.e., $\Delta(x) = \Delta(x, 0^n)$. By default, logarithms are taken to base 2. For real $p \in [0,1]$ we let $H_2(p) := -p\log(p) - (1-p)\log(1-p)$ denote the binary entropy function where $0 \log 0$ is taken to be 0. The inverse of the binary entropy function, $H_2^{-1} : [0,1] \to [0, \frac{1}{2}]$, maps $y \in [0,1]$ to the unique $x \in [0, \frac{1}{2}]$ for which $H_2(x) = y$. It is well known (cf. [44, Chapter 3]) that for every constant $\delta \in (0, 1/2)$

$$2^{nH_2(\delta)-o(n)} \leq \binom{n}{\delta n} \quad \text{and} \quad \sum_{i=1}^{\delta n} \binom{n}{i} \leq 2^{nH_2(\delta)}. \tag{2.1}$$

We also use the following approximation taken from [24, Theorem 2.2]:

$$\frac{x}{2\log(6/x)} \leq H_2^{-1}(x) \leq \frac{x}{\log(1/x)}. \tag{2.2}$$

A function $\epsilon(\cdot)$ is said to be negligible if $\epsilon(k) < k^{-c}$ for any constant $c > 0$ and sufficiently large k. We will sometimes use $\mathrm{neg}(\cdot)$ to denote an unspecified negligible function. The statistical distance

between two probability distributions X and Y, denoted $\mathsf{SD}(X;Y)$, is defined as the maximum, over all functions A, of the distinguishing advantage $|\Pr[A(X)=1]-\Pr[A(Y)=1]|$. A pair of distribution ensembles $X=\{X_k\}$ and $Y=\{Y_k\}$ is *statistically indistinguishable* if $\mathsf{SD}(X_k;Y_k)\leq \mathrm{neg}(k)$.

*Locality and Degree. Let $f:\{0,1\}^k\to\{0,1\}^m$ be a function. We say that the i-th output variable y_i *depends* on the j-th input variable x_j (or equivalently, x_j *affects* the output y_i) if there exists a pair of input strings which differ only on the j-th location whose images differ on the i-th location. The locality of an output variable (resp., input variable) is the number of input variables on which it depends (resp., the number of output variables which it affects). We say that an output has degree d if it can be expressed as a multivariate polynomial of degree d in the inputs over the binary field $\mathbb{Z}_2$. The locality of an output variable trivially upper bounds its degree. The output locality (resp., degree) of f is the maximum output locality (resp., degree) over all outputs of f. Similarly, the input locality of f is the maximal input locality over all inputs of f.

Definition 2.1 (Collision-Resistant Hash Functions)**.** A collection of functions

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

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

- (Shrinkage) The output length is smaller than the input length: $m(k)<k$ for every k.
- (Efficient evaluation and sampling) There exists a pair of efficient algorithms: (a) an *evaluation algorithm* H which given $(z\in\{0,1\}^s, x\in\{0,1\}^k)$ outputs $h_z(x)$; and (b) a key-sampling algorithm $\mathcal{K}$ which given 1^k samples an index $z\in\{0,1\}^{s(k)}$.
- (Collision resistance) For every probabilistic polynomial-time adversary Adv it holds that

$$\Pr_{z\xleftarrow{R}\mathcal{K}(1^k)}[\mathsf{Adv}(z)=(x,x') \text{ s.t. } x'\neq x \text{ and } h_z(x)=h_z(x')] \tag{2.3}$$

  is negligible in k.

The *shrinking factor* of $\mathcal{H}$ is the ratio m/k. We say that $\mathcal{H}$ is *linearly shrinking* if m/k is upper-bounded by a constant $c<1$. Similarly, $\mathcal{H}$ is *polynomially shrinking* if $m/k<1/k^c$ for some constant $c\in(0,1)$.

The weaker variant of *universal one-way hash function* (UOWHF) [63] is defined by relaxing the third item (collision resistance) with the following requirement (also known as *target collision resistance* [11]):

- (Target collision resistance) For every pair of probabilistic polynomial-time adversaries $\mathsf{Adv}=(\mathsf{Adv}_1,\mathsf{Adv}_2)$ it holds that

$$\Pr_{\substack{(x,r)\xleftarrow{R}\mathsf{Adv}_1(1^k)\\ z\xleftarrow{R}\mathcal{K}(1^k)}}[\mathsf{Adv}_2(z,x,r)=x' \text{ s.t. } x'\neq x \text{ and } h_z(x)=h_z(x')]\leq \mathrm{neg}(k).$$

That is, first the adversary Adv_1 specifies a target string x and a state information r, then a random hash function h_z is selected, and then Adv_2 tries to form a collision x' with x under h_z.

Remark 2.2 (Measuring efficiency)**.** When saying that a collection $\mathcal{H}$ of hash functions enjoys some level of efficiency we refer to the complexity of *every fixed* function h_z in the collection $\mathcal{H}$. For example, $\mathcal{H}$ has constant output locality of d if every function $h_z \in \mathcal{H}$ has output locality of at most d. A stronger form of efficiency guarantees that given the index z and the input x the function $H(z,x) = h_z(x)$ has the required level of efficiency (e.g., constant locality). Since the index z is selected once and for all we adopt the former (weaker) variant as our default notion. However, some of our constructions also guarantee the stronger form of efficiency.

Remark 2.3 (Public coins)**.** Our constructions are all in the "public-coin" setting [49], and so they remain secure even if the adversary gets the coins used to sample the index of the collection.

2.1 Randomized Encoding of Functions

Roughly speaking, a *randomized encoding* [51, 1] of a function $f(x)$ is a randomized mapping $\hat{f}(x;r)$ such that for every input x the output distribution $\hat{f}(x;r)$ (induced by a random choice of r) depends only on the output of $f(x)$. Throughout the paper we employ *perfect randomized encoding* as defined below.

Definition 2.4 (Perfect Randomized Encoding)**.** Let $f : \{0,1\}^n \to \{0,1\}^m$ be a function. We say that a function $\hat{f} : \{0,1\}^n \times \{0,1\}^\rho \to \{0,1\}^s$ is a *perfect randomized encoding* (PRE) of f if there exists a deterministic decoding algorithm C and a randomized simulator S which satisfy the following:

- (Perfect correctness) For every input $x \in \{0,1\}^n$ and $r \in \{0,1\}^\rho$, it holds that $C(\hat{f}(x;r)) = f(x)$.
- (Perfect privacy) For every $x \in \{0,1\}^n$, the distribution $\hat{f}(x;r)$, induced by a uniform choice of $r \stackrel{R}{\leftarrow} \{0,1\}^\rho$, is identical to the distribution $S(f(x))$.
- (Balanced simulation) The distribution $S(y)$ induced by choosing $y \stackrel{R}{\leftarrow} \{0,1\}^m$ is identical to the uniform distribution over $\{0,1\}^s$.
- (Length preserving) The difference between the output length and the total input length of the encoding $s - (n + \rho)$ is equal to the difference $m - n$ between the output length and the input length of f.

We refer to the second input of $\hat{f}$ as its *random input* and to ρ and s as the *randomness complexity* and *output complexity* of $\hat{f}$, respectively.

The definition naturally extends to collections of functions $\mathcal{F} = \left\{ f_z : \{0,1\}^{n(z)} \to \{0,1\}^{m(z)} \right\}_{z \in \{0,1\}^*}$. In particular, we say that $\hat{\mathcal{F}} = \left\{ \hat{f}_z : \{0,1\}^{n(z)} \times \{0,1\}^{\rho(z)} \to \{0,1\}^{s(z)} \right\}_{z \in \{0,1\}^*}$ perfectly encodes $\mathcal{F}$ if for every z, $\hat{f}_z$ perfectly encodes f_z. Furthermore, we always assume that the encoding is uniform in the sense that there exists a polynomial-time algorithm which given z outputs a description (say as a boolean circuit) of the encoding $\hat{f}_z$, its decoder C_z and its simulator S_z.

In [1] it is shown that a PRE of a CRH is also CRH.

Lemma 2.5 ([1, Lemma 7.2])**.** *If $\mathcal{H} = \left\{ h_z : \{0,1\}^k \to \{0,1\}^m \right\}$ is a CRH then its perfect encoding $\hat{\mathcal{H}} = \left\{ \hat{h}_z : \{0,1\}^k \times \{0,1\}^\rho \to \{0,1\}^s \right\}$ is also a CRH, where $\hat{\mathcal{H}}$ is viewed as a collection of single-input functions (of input length $k + \rho$) which uses the key-sampling algorithm of $\mathcal{H}$.*

Observe that if the collection $\mathcal{H}$ has linear-shrinkage and the perfect encoding $\hat{\mathcal{H}}$ has randomness complexity of $O(k)$, then the collection $\hat{\mathcal{H}}$ also has linear shrinkage.

The following proposition follows from [1, Section 4].

Proposition 2.6. Let $f : \{0,1\}^n \to \{0,1\}^\ell$ be a function that each of its outputs can be computed by a formula of size at most t. Then, f can be encoded by a PRE $\hat{f}$ with degree 3, output locality 4, randomness/output complexity of $\ell \cdot \mathrm{poly}(t)$ and input locality of at most $c \cdot \mathrm{poly}(t)$ where c is the input locality of f. In particular, if f has constant output locality then the randomness complexity of $\hat{f}$ is $O(\ell)$, and if, in addition, f has constant input locality, then $\hat{f}$ also has constant input locality.

We will also need two standard closure properties of PREs. First, just like in the case of string-encodings, if we take an encoding $\hat{f}$ of f, and re-encode it by $\hat{\hat{f}}$, then the resulting encoding also encodes the original function f. Second, given an encoding $\hat{f}(y;r)$ of $f(y)$ we can encode a function of the form $f(g(x))$ by encoding the outer function and substituting y with $g(x)$, i.e., $\hat{f}(g(x);r)$. We summarize these properties via the following lemmas (taken from [1, Lemma 4.11] and [2, Fact 3.1]).

Lemma 2.7 (Composition lemma)**.** *Suppose that $g(x;r_g)$ is a PRE of $f(x)$ and $h((x,r_g);r_h)$ is a PRE of $g((x,r_g))$ (viewed as a single-argument function). Then, the function $\hat{f}(x;(r_g,r_h)) \stackrel{\Delta}{=} h((x,r_g);r_h)$ is a PRE of f.*

Lemma 2.8 (Substitution lemma)**.** *Suppose that the function $\hat{f}(x;r)$ is a PRE of $f(x)$. Let $h(z)$ be a function of the form $f(g(z))$ where $z \in \{0,1\}^k$ and $g : \{0,1\}^k \to \{0,1\}^n$. Then, the function $\hat{h}(z;r) \stackrel{\Delta}{=} \hat{f}(g(z);r)$ is a PRE of h.*

3 Our Assumptions

3.1 The Binary Shortest Vector Problem

In the following the term *matrix sampler* refers to an algorithm $\mathcal{M}$ which, given 1^n, samples $m(n) \times n$ binary matrix for some integer-valued function $m(n)$.

Definition 3.1 (binary SVP)**.** For a weight parameter $\delta(n) : \mathbb{N} \to (0, 1/2)$, and an efficient sampler $\mathcal{M}(1^n)$ which samples $m(n) \times n$ binary matrices, the $(\mathcal{M}, \delta)$-bSVP assumption asserts that for every efficient algorithm Adv the probability

$$\Pr_{\mathbf{M} \stackrel{R}{\leftarrow} \mathcal{M}(1^n)} [\mathsf{Adv}(\mathbf{M}) = \mathbf{x} \text{ such that } \mathbf{x} \neq \mathbf{0}, \mathbf{M}\mathbf{x} = \mathbf{0} \text{ and } \Delta(\mathbf{x}) \leq \delta]$$

is negligible in n. If $m(n)/n \leq \alpha(n)$, then we refer to the distribution sampled by $\mathcal{M}$ as (α, δ)-bSVP *hard distribution*.

Using a coding-theoretic terminology, we can think of $\mathcal{M}$ as specifying an ensemble of binary linear codes which are represented by their $m \times n$ parity-check matrices. We will always assume that, except with negligible probability, the rows of $M \stackrel{R}{\leftarrow} \mathcal{M}(1^n)$ are linearly independent and so the code has rate of $1 - m/n$. The binary SVP assumption asserts that it is hard to find a short codeword (of weight at most δ) in a random member of this ensemble. For the purpose of constructing hash functions, we will choose $\delta(n)$ such that a code sampled from $\mathcal{M}(1^n)$ is likely to contain (exponentially) many codewords of weight at most $\delta(n)$ (but such light codewords capture only exponentially-small fraction of all codewords). Intuitively, this setting corresponds to the

list-decoding regime of the code.[4] We mention that in the worst case, it is NP-hard to compute the distance of a linear code [72] or even to approximate it by a constant factor [33]. Currently, the best known algorithms run in exponential time (cf. [71, 31, 8, 36, 14, 10]). Let us present the main distributions used in the paper.

3.1.1 The Random Linear Code Ensemble

The ensemble of *random linear codes* is probably the most natural choice for bSVP. Formally, for a length parameter $\alpha(n) : \mathbb{N} \to (0,1)$ and weight parameter $\delta(n)$, we let (α,δ)-bSVP denote the $(\mathcal{M},\delta)$-bSVP assumption where $\mathcal{M}(1^n)$ uniformly samples a matrix from $\mathbb{Z}_2^{\lceil \alpha(n)\cdot n\rceil \times n}$. It is well known that a random linear code of rate $R = (1-\alpha)$ achieves the Gilbert-Varshamov bound (cf. [44]). Specifically, for any constants δ, α for which $\delta < H_2^{-1}(\alpha)$, a uniformly chosen matrix $\mathbf{M} \xleftarrow{R} \mathbb{Z}_2^{\alpha n \times n}$ has no codewords of weight less than δ (except with exponentially small probability $\exp(-\Omega(n))$). Accordingly, we will be interested in the regime where $\delta > H_2^{-1}(\alpha)$. For this regime we make the following simple observations. For simplicity, we restrict our attention to the case where α and δ are constants which do not depend on n.

Observation 3.2 (Attack based on linear algebra)**.** *If $\delta > \alpha/2$ the (α,δ)-bSVP assumption does not hold. In particular, there is a polynomial-time algorithm that solves the problem with probability $\frac{1}{2}$.*

Proof. Given $\mathbf{M}$, we can always find a set S of $m(n) = \lceil \alpha n \rceil$ linearly-independent columns which span the column space (since the matrix has only $m(n)$ rows). We can therefore find a solution $\mathbf{x}$ of weight αn to the original system by letting $\mathbf{x}_S$ be the unique vector in the kernel of the restricted matrix $\mathbf{M}_S$, and by letting $\mathbf{x}_{[n]\setminus S}$ be the all zero vector.

To do better, we choose S so that the random variable M_S (induced by M) is a random full rank square matrix (e.g., by choosing the lexicographically-first S). In this case, the (unique) vector $\mathbf{x}_S$ in $\ker(\mathbf{M}_S)$ is uniformly distributed over $\mathbb{Z}_2^{\alpha n}$ and so with probability $\frac{1}{2}$ its relative weight is at most $\frac{1}{2}$. It follows that, with probability $\frac{1}{2}$, the algorithm outputs a vector in the kernel of $\mathbf{M}$ whose Hamming weight is at most $\alpha/2$. ∎

We do not know of any polynomial-time attack for the case $\delta < \alpha/2$.

Observation 3.3. *If (α,δ)-bSVP holds for $\delta > H_2^{-1}(\alpha)$ then one-way functions exist.*

Proof. Let $m(n) = \lceil \alpha n \rceil$. Consider the algorithm S which samples a random $\mathbf{x} \in \mathbb{Z}_2^n$ of weight $\lfloor \delta n \rfloor$ and then samples a random matrix $\mathbf{M} \in \mathbb{Z}_2^{m(n)\times n}$ subject to $\mathbf{M}\cdot\mathbf{x} = \mathbf{0}$. We claim that the mapping f that takes the random coins of the algorithm and outputs $\mathbf{M}$ is one-way. Indeed, assume, towards a contradiction, that the mapping can be inverted efficiently by an adversary Adv with probability ϵ, then, we can break (α,δ)-bSVP by applying Adv on $\mathbf{M} \xleftarrow{R} \mathbb{Z}_2^{m(n)\times n}$, get r, and apply the sampler in the forward direction to recover a solution $\mathbf{x}$.

To analyze the success probability it suffices to show that the distribution sampled by S (on which Adv is promised to succeed) is statistically close to the uniform distribution over $\mathbb{Z}_2^{m(n)\times n}$. Indeed, this follows by noting that (1) S samples the uniform distribution over all matrices whose kernel contains a vector of relative weight δ; and (2) The probability that a uniformly chosen parity-check matrix $\mathbf{M} \xleftarrow{R} \mathbb{Z}_2^{m(n)\times n}$ will not have a vector of weight δ in its kernel is exponentially small (cf. [9]). This completes the proof. ∎

[4] In fact, we show that, over random linear codes, a variant of the list-decoding problem reduces to bSVP (see Lemma 6.1).

We will later show (Theorem 4.4) that hardness for $\delta > 2H_2^{-1}(\alpha)$ implies the existence of collision-resistance hash functions. Due to the above attack, this means that the weight parameter δ should be in the interval

$$(2H_2^{-1}(\alpha), \alpha/2).$$

Plugging in the approximation from Eq. (2.2), and letting $\alpha = 2^{-k}$, we conclude that δ should live in the interval

$$(2^{-k+1}/k, 2^{-k-1}),$$

so the ratio between the upper-bound and the lower-bound grows when $\alpha = 2^{-k}$ decreases.

3.1.2 Random LDPC Ensemble

Instead of taking a uniformly-chosen parity-check matrix, one can use an ensemble of Low-Density Parity-Check Codes (LDPC) [37]. Concretely, for constant $\alpha \in (0,1)$ and constant $d \in \mathbb{N}$ for which $c = \alpha d$ is an integer, we let (d, α, δ)-bSVP denote the $(\mathcal{M}_{\alpha,d}, \delta) - \mathsf{bSVP}$ assumption where $\mathcal{M}_{\alpha,d}(1^n)$ samples a uniformly chosen $\alpha \cdot n \times n$ matrix subject to the constraint that each column contains exactly c ones and each row contains exactly d ones.[5]

This ensemble of codes is well studied in the coding theory literature. Most notably, it is known that, for any fixed α and $d > 2$, a typical code in the ensemble can be efficiently decoded in the presence of constant noise rate of $p(\alpha, d) > 0$ (say over the binary symmetric channel) [37]. So in some regime of parameters, the unique decoding problem over LDPCs can be solved efficiently. Interestingly, for a fixed α, larger sparsity d reduces the noise rate $p(\alpha, d)$ for which known efficient decoding techniques work [23], whereas, combinatorially, when d grows the code becomes better and approaches the performance of a random linear code [37, 57].

Several methods for finding codewords of minimal weight in LDPC's have been proposed (see [48, 50, 74, 53, 32] and references therein). It is typically unknown how to analyze the complexity of these heuristics but an experimental study seems to suggest that the complexity grows exponentially with the distance of the code which is linear in the block length n (see also the discussion in [32]).

Overall, one may conjecture that when sparsity grows the intractability of binary SVP over LDPC codes "approaches" the intractability of SVP over the random linear code ensemble. Specifically, it seems plausible that for some constant α and every $\delta < \alpha/2$ there exists a (sufficiently large) constant d for which (d, α, δ)-bSVP holds.

3.2 Multivariate Quadratic Assumptions

We first define the multivariate quadratic (MQ) assumption that we use in this work.

Definition 3.4. Let $\mathcal{D} = \{\mathcal{D}_{n(\lambda),m(\lambda),p(\lambda)}\}_{\lambda\in\mathbb{N}}$ be an ensemble of probability distributions that output a sequence of $m = m(\lambda)$ *upper triangular* matrices $\mathbf{Q}_1, \ldots, \mathbf{Q}_m \in \mathbb{Z}_p^{n\times n}$ (where we will write p for $p(\lambda)$ and n for $n(\lambda)$ from now on), and m vectors $\mathbf{L}_1, \ldots, \mathbf{L}_m \in \mathbb{Z}_p^n$. The $\mathcal{D}$-multivariate quadratic assumption (which we will refer to simply as the MQ assumption, when the parameter $\mathcal{D}$ is obvious from the context) states that it is computationally hard to find a *non-zero* solution to a given set of m quadratic equations

$$\left\{ q_i(\mathbf{x}) \triangleq \mathbf{x}^T\mathbf{Q}_i\mathbf{x} + \mathbf{L}_i^T\mathbf{x} \triangleq \sum_{j,k\in[n]} q_{i,j,k}x_jx_k + \sum_{j\in[n]} \ell_{i,j}x_j = 0 \pmod p \right\}_{i\in[m]}$$

[5] We implicitly restrict our attention to n's for which $(c/d)\cdot n$ is an integer, and, correspondingly, assume hardness only for these input lengths. Nevertheless, since this set of inputs is sufficiently dense, we can derive collision resistant hash functions for all input lengths via standard padding.

where $q_{i,j,k}$ are the (j,k)-th entries of the matrix $\mathbf{Q}_i$ and $\ell_{i,j}$ are the j-th entries of the vector $\mathbf{L}_i$.

3.2.1 Previous Work on the MQ Problem

It is well-known that the multivariate quadratic (MQ) problem is NP-hard in the worst case [38]. To the best of our knowledge, the best algorithms for the *random* MQ problem with $m = O(n)$ run in $2^{\Omega(n)}$ time. Kipnis, Patarin and Goubin [55] showed that a random instance of MQ can be solved in polynomial time if $m(n) = O(\sqrt{n})$. Kipnis and Shamir [56] showed that the MQ problem can be solved in polynomial time when $m(n) = \Omega(n^2)$. The hardness of the MQ function for a randomly chosen instance is not known to follow from any well-studied intractability assumption.

Regarding cryptographic usefulness, it is easy to see that the average-case hardness of the MQ problem immediately gives us a one-way function. In the regime where $m > n$, the same assumption also gives us a pseudorandom generator [13].

The early work using the MQ assumption, starting from Matsumoto and Imai [59], focused on the (much) harder task of constructing public-key encryption schemes. The hardness of the MQ problem was necessary but not sufficient for the semantic security of their encryption scheme. Indeed, their proposal was attacked [64, 56] and fixed many times. However, none of the attacks break the MQ assumption, but rather were the result of the additional structure introduced into the assumption to obtain public-key cryptography. We do not go into the details of this long line of work as public-key cryptography is not the focus of our paper. However, a reader interested in the history of this early work is referred to Christopher Wolf's Ph.D. thesis [73].

Moving on to hashing, the topic of this paper, Aumasson and Meier [5] showed that a *random* and *sparse* degree-2 function is not collision-resistant. Ding and Yang [30] claim that the sparsity condition can be removed, namely that a random degree-2 function with compression level $m(n) = n/2$ is not collision-resistant; however, no formal proof of this claim was provided. (Jumping ahead, we will show that indeed, their intuition was correct and that one can find in polynomial time collisions in the random MQ function with *any* non-trivial compression.) Ding and Yang [30] also conjectured that a random degree-3 function with the same compression level is a CRH. Billet, Robshaw and Peyrin [15] observed that given the difference between a possible colliding inputs of degree-2 function, the collision can be found in polynomial time. This method was first presented by Patarin in [64].

We are not aware of any results on universal one-way hashing from MQ-type assumptions.

4 Hash Functions from the Binary SVP Assumption

4.1 A General Template

We show how to construct CRH based on the bSVP assumption. Our CRH is keyed with a matrix $\mathbf{M} \in \mathbb{Z}_2^{m \times n}$, it first takes an input $\mathbf{x} \in \{0,1\}^k$ and expand it into an n-bit vector $\mathbf{y}$ via some preprocessing mapping Expand, and then compresses $\mathbf{y}$ to an m-bit vector $\mathbf{z}$ by computing $\mathbf{My}$. Formally, the construction has the following structure.

Construction 4.1. Let $n = n(k)$ and $m = m(k)$ be integer valued functions. For a mapping $\mathsf{Expand} : \{0,1\}^k \to \{0,1\}^n$, and a matrix-sampler $\mathcal{M}(1^n)$ which samples $m \times n$ matrices, we define the collection of functions $\mathcal{H}_{k,m} = \{h_{\mathbf{M}} : \{0,1\}^k \to \{0,1\}^m \ : \ \mathbf{M} \in \mathbb{Z}_2^{m \times n}\}$ where

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

and the key-sampling algorithm $\mathcal{K}(1^k)$ outputs $\mathbf{M} \xleftarrow{R} \mathcal{M}(1^n)$.

We say that a function $\mathsf{Expand} : \{0,1\}^k \to \{0,1\}^n$ is (b,β)-*expanding* if (1) the function is injective; (2) the expansion factor n/k is at most b; and (3) the function outputs strings whose relative Hamming weight is at most β.

Lemma 4.2. *Suppose that Construction 4.1 is instantiated with a matrix sampler $\mathcal{M}(1^n)$ which samples an (α,δ)-*bSVP *hard distribution and with a (b,β)-expanding algorithm* Expand *where $b\alpha < 1$ and $2\beta \leq \delta$. Then, the resulting collection $\mathcal{H}$ is a collision-resistance hash function with shrinkage factor of $b\alpha$.*

Proof. First observe that since $b\alpha < 1$ the function $h_{\mathbf{M}}$ is shrinking, as required. Next, we show that a collision finder Adv can be used to find a short vector in the kernel of $\mathbf{M}$. Assume, towards a contradiction, that there exists an efficient collision-finder Adv that given an $m(k) \times n(k)$ matrix $\mathbf{M} \xleftarrow{R} \mathcal{K}(1^k)$ outputs a collision $\mathbf{x} \neq \mathbf{x}' \in \mathbb{Z}_2^k$ with non-negligible probability $\epsilon(k)$. We show that the vector $\mathbf{y} = \mathsf{Expand}(\mathbf{x}) \oplus \mathsf{Expand}(\mathbf{x}')$ is (1) non-zero vector (2) it has relative weight of at most δ, and (3) it is in $\ker(\mathbf{M})$. Indeed, (1) follows since Expand is injective, (2) follows since the image of Expand contains only strings whose relative Hamming is at most β and so the vector $\mathbf{y}$ has Hamming weight of at most $2\beta \leq \delta$. Finally, since the pair $(\mathbf{x},\mathbf{x}')$ forms a collision, it holds that $\mathbf{M} \cdot \mathsf{Expand}(\mathbf{x}) = \mathbf{M} \cdot \mathsf{Expand}(\mathbf{x}')$ and therefore (3) follows as well. ∎

In the following we show that one can construct (b,β) expanding algorithm with constant input and output locality (and therefore also with constant degree and linear circuit size). The first part of the lemma optimizes the parameters b and β (and achieves large locality parameters) and the second part optimizes locality (at the expense of a looser relation between b and β).

Lemma 4.3. *For every constant $\beta \in (0,1/2)$ (weight upper-bound) the following holds.*

1. *For every $b > 1/H_2(\beta)$ there exists an efficiently computable mapping* $\mathsf{Expand} : \{0,1\}^k \to \{0,1\}^{n(k)}$ *which is (b,β)-expanding for all sufficiently large k's and has constant input locality c and constant output locality d where c and d depend on β and b.*
2. *If β is a power of $1/2$ then there exists an efficiently computable $(b = \frac{1}{\beta\log(1/\beta)},\beta)$-expanding mapping* $\mathsf{Expand}' : \{0,1\}^k \to \{0,1\}^{n(k)}$ *with output locality of $\log(1/\beta)$ and input locality of $1/\beta$.*

Note that $b > 1/H_2(\beta)$ is a necessary requirement (otherwise by Eq. (2.1) the image of the mapping contains less than 2^k strings and so it cannot be injective). Therefore, the first part of the lemma achieves an optimal dependency between b and β.

Proof. (1) Without the locality constraint, the algorithm $\mathsf{Expand}_{\beta,b}$ can be easily implemented. Indeed, given an input $x \in \{0,1\}^k$ interpreted as an integer in $[1,2^k]$, we can output the lexicographically x-th n-bit string of weight w efficiently by computing the output $y \in \{0,1\}^n$ in a bit-by-bit manner. (E.g., compute the number $T = \binom{n-1}{w}$ of n-bit word of weight w that begin with zero, set y_1 to zero if $x < T$ and to 1 otherwise, and continue recursively with the other bits.) Setting $n = \lfloor kb \rfloor$ and $w = \lfloor \beta n \rfloor$, and recalling that, by Eq. (2.1), for $b > 1/H_2(\beta)$ and sufficiently large k, there are more than 2^k strings of length n and weight w, we conclude that the mapping is injective.

To get low locality choose sufficiently large constants c and d such that $1/H_2(\beta) < c/d < b$ and such that $\mathsf{Expand}_{\beta,c/d}$ is injective for inputs of length d. Now partition your k-bit input into k/d-blocks of size d each. Apply $\mathsf{Expand}_{\beta,c/d}$ to each such block of inputs and generate an output block of length at most c. Output the concatenation of all k/d output blocks. By definition, the mapping has the desired expansion and locality. The (b,β)-expansion property is inherited from the original $\mathsf{Expand}_{\beta,c/d}$ algorithm.

(2) The procedure Expand' works as follows: It splits the k input bits into blocks of size $\log(1/\beta)$ bits. For each block $\mathbf{z} \in \{0,1\}^{\log(1/\beta)}$, compute a block $\mathbf{z}' \in \{0,1\}^{1/\beta}$ which is 1 in exactly the $\mathbf{z}^{th}$ location. It is easy to check that the output length n is $\frac{k}{\beta \log(1/\beta)}$ and that its Hamming weight is $\frac{k}{\log(1/\beta)} = \beta n$, as required. It is also clearly injective and can be computed with output locality of $\log(1/\beta)$ and input locality of $1/\beta$. ■

4.2 Degree-3 CRH

Based on Lemmas 4.2 and 4.3, we prove the following theorem.

Theorem 4.4. *Suppose that there exist constants $\delta \in (0, 1/4), \alpha \in (0,1)$ with $\delta > 2H_2^{-1}(\alpha)$ and an efficient matrix sampler $\mathcal{M}$ which samples some (α, δ)-bSVP hard distribution. Then, there exists a linearly shrinking CRH with constant degree.*

Proof. Fix δ and α and take $\beta = \delta/2$ and $b \in (1/H_2(\beta), 1/\alpha)$ (the interval is non-empty due to the requirement $\delta > 2H_2^{-1}(\alpha)$). Instantiate Construction 4.1 with the matrix sampler $\mathcal{M}$ and the (b, β)-expanding mapping Expand promised in item 1 of Lemma 4.3. Then, by Lemma 4.2, we derive a linearly-shrinking CRH with constant degree. ■

Our next goal is to turn the above construction into a degree-3 CRH with linear shrinkage. To this end, we show that the CRH constructed in Theorem 4.4 admits a degree-3 perfect randomized encoding that uses only linear amount ($O(k)$) of random bits.

Lemma 4.5. *Let $h(\mathbf{x})$ be a function of the form $\mathbf{M} \cdot g(\mathbf{x})$ where $g : \mathbb{Z}_2^k \to \mathbb{Z}_2^n$ is an NC^0 function and $\mathbf{M}$ is an $m \times n$ matrix of full rank with $m < n$. Then, h can be perfectly encoded by a degree-3 function with randomness complexity of $O(n)$.*

Recall that any function can be perfectly encoded by a degree-3 encoding (Proposition 2.6), however the randomness complexity of the best known general transformations grows polynomially with the total size of the formulas (or branching programs) that compute the output bits of the encoded function. Such a blowup will give us only sub-linear shrinkage. The lemma bypasses this problem by introducing a new randomness-efficient degree-3 encoding which is tailored to the structure of the CRH constructed in Theorem 4.4.

Proof. Let $\ell = n - m$ and take $\mathbf{L} \in \mathbb{Z}_2^{n \times \ell}$ to be a matrix which spans the kernel of $\mathbf{M}$. We begin by observing that the function h is perfectly encoded by the function

$$\hat{h}(\mathbf{x}; \mathbf{r}) = g(\mathbf{x}) + \mathbf{L}\mathbf{r},$$

where $\mathbf{r} \in \mathbb{Z}_2^\ell$. Indeed, given $\mathbf{z} = \hat{h}(\mathbf{x}; \mathbf{r})$ we can decode $h(\mathbf{x})$ by computing $\mathbf{M} \cdot \mathbf{z}$. On the other direction, given $\mathbf{y} = h(\mathbf{x})$ we can perfectly simulate $\mathbf{z} = \hat{h}(\mathbf{x}; \mathbf{r})$ by sampling a random preimage of $\mathbf{y}$ under $\mathbf{M}$. Since $\mathbf{M}$ has full rank, the resulting simulator is balanced. Finally, the encoding is stretch preserving since the output-input difference $n-(k+\ell)$ of $\hat{h}$ equals to $m-k$, the output-input difference of h.

Next consider the NC^0 function $f(\mathbf{x}, \mathbf{y})$ which takes $\mathbf{x} \in \mathbb{Z}_2^k$ and $\mathbf{y} \in \mathbb{Z}_2^n$ and outputs $g(\mathbf{x}) + \mathbf{y}$. By Proposition 2.6, f can be perfectly encoded by a degree-3 encoding $\hat{f}(\mathbf{x}, \mathbf{y}; r')$ with randomness complexity of $O(ns^2) = O(n)$.

Finally, observe that the function $\hat{h}(\mathbf{x}; \mathbf{r})$ can be written as $f(\mathbf{x}, \mathbf{L}\mathbf{r})$. Therefore, by the substitution lemma (Lemma 2.8), the function $e(\mathbf{x}, \mathbf{r}; r') \stackrel{\Delta}{=} \hat{f}(\mathbf{x}, \mathbf{L}\mathbf{r}; r')$ perfectly encodes the function $\hat{h}(\mathbf{x}, \mathbf{r})$. Hence, by the composition lemma (Lemma 2.7), the function $e(\mathbf{x}; \mathbf{r}, r')$ perfectly encodes h. Noting that e has degree 3 and randomness complexity of $O(n + \ell) = O(n)$, the lemma follows. ■

Recall that we always assume that a hard-bSVP-distribution puts all but a negligible fraction of its mass on matrices whose columns are linearly independent. Hence, we can apply Lemma 4.5 to the CRH constructed in Theorem 4.4, and derive, by Lemma 2.5, the following improved version of Theorem 4.4.

Theorem 4.6. *Under the hypothesis of Theorem 4.4, there exists a linearly shrinking CRH with degree 3.*

Instantiating bSVP with the random linear code ensemble, we derive the following corollary.

Corollary 4.7. *Suppose that there exist constants $\delta \in (0, 1/4), \alpha \in (0, 1)$ which satisfy $\delta > 2H_2^{-1}(\alpha)$ for which $(\alpha, \delta) -$ bSVP holds. Then, there exists a degree-3 linearly-shrinking CRH.*

Corollary 4.7 serves as a feasibility result for the existence of degree-3 CRH with linear shrinkage. This statement attempts to optimize the parameters of the underlying intractability assumption (making it as plausible as possible) and the degree, but yields a poor (constant) shrinkage factor. By iterating the CRH a sufficiently large (constant) number of times, we can reduce the shrinkage factor to an arbitrary constant ϵ at the expense of increasing the degree to a large constant $d = d(\epsilon)$. Alternatively, we can get a better tradeoff between the degree and the shrinkage factor by strengthening the assumption as follows.

Proposition 4.8. Suppose that $(\alpha, \delta) -$ bSVP holds for every constants $\delta \in (0, 1/4), \alpha \in (0, 1)$ which satisfy $\delta < \alpha/2$. Then, for every constants $d > 4$ and $\gamma > 4/d$ there exists a degree-d CRH with shrinkage factor of γ.

For example, we can get a degree 5 CRH with shrinkage-factor of 0.81 or degree-8 CRH with shrinkage factor of 0.51.

Proof. Let $\beta = 2^{-d}$ and recall that item 2 of Lemma 4.3 provides a (b, β)-expanding mapping Expand′ with $b = \frac{1}{\beta \log(1/\beta)} = 2^d/d$ and output locality (and therefore also degree) of $\log(1/\beta) = d$. Let $\delta = 2\beta = 2^{-d+1}$ and $\alpha = \gamma/b$. Since $\gamma > 4/d$, it follows that $\delta < \alpha/2$, which, according to our assumption, implies that $(\alpha, \delta) -$ bSVP holds. By plugging Expand′ and the matrix-sampler that samples uniform $\alpha n \times n$ matrices into Construction 4.1, we get, by Lemma 4.2, a degree-d CRH with shrinkage factor of γ. The proposition follows. ■

Remark 4.9. Recall that we measure the degree of a collection of functions $H = \{h_z\}$ as the maximal degree of each function in the collection and ignore the degree of the evaluation algorithm H which maps the collection key z and the input x to $h_z(x)$. (See Remark 2.2.) Nevertheless, it is not hard to see that all the constructions of this section admit an evaluation algorithm of constant degree. (In fact, the degree is $d + 1$ where d is the degree of h_z).

4.3 Locally-Computable CRH

Our next goal is to construct CRH's with constant output and input locality. To this end, we instantiate bSVP with the LDPC ensemble.

Theorem 4.10. *Suppose that there exist constants $\delta \in (0, 1/4), \alpha \in (0, 1)$ with $\delta > 2H_2^{-1}(\alpha)$ and a constant $d \in \mathbb{N}$ for which $(d, \alpha, \delta) -$ bSVP holds. Then, there exists a linearly-shrinking CRH with constant input locality and constant output locality. Moreover, one can reduce the output locality to 4 (while keeping the shrinkage linear and the input locality constant).*

Proof. Fix δ and α and take $\beta = \delta/2$ and $b \in (1/H_2(\beta), 1/\alpha)$. Let Expand be the (b, β)-expanding mapping promised in item 1 of Lemma 4.3 which has input locality of c' and output locality d'. Instantiate Construction 4.1 with Expand and with the LDPC matrix sampler $\mathcal{M}$ which samples a uniformly chosen $\alpha \cdot n \times n$ matrix subject to the constraint that each column contains exactly $c = \alpha d$ ones and each row contains exactly d ones. Then, by Lemma 4.2, we derive CRH $\mathcal{H} = \{h_{\mathbf{M}}\}$ with linear shrinkage, output locality of $D = d \cdot d'$ and input locality of $C = c \cdot c'$. This proves the first part of the theorem.

For the second part, take the CRH $\mathcal{H} = \left\{h_z : \{0,1\}^k \to \{0,1\}^m\right\}$ constructed in the first part of the theorem, and apply the 4-local perfect encoding promised in Proposition 2.6. The resulting collection $\hat{\mathcal{H}}$ has the required syntactic properties, and, by Lemma 2.5, it forms a CRH. ■

Since any NC^0 function can be computed by a circuit of linear size, Theorem 4.4 yields a linear-time computable CRH with linear-shrinkage. Such a function can be turned into a linear-time computable CRH with arbitrary polynomial-stretch (using a hash-tree), we therefore derive the following corollary.

Corollary 4.11. *Under the assumption of Theorem 4.10, for every constant $c < 1$ there exists a CRH $\mathcal{H} = \left\{h_z : \{0,1\}^k \to \{0,1\}^{k^c}\right\}$ which can be computed by a circuit of size $O(k)$.*

5 Degree-2 Hash Functions

In Section 5.1, we construct a universal one-way hash function family under the MQ assumption (see Definition 3.4 for the definition of the $\mathcal{D}_{n,m,p}$-MQ assumption). We then turn our attention to collision-resistance. In Section 5.2, we show that a natural family of uniformly random quadratic functions is *not* collision-resistant, by showing an explicit polynomial-time attack. An intriguing question left open by our work is the existence of a degree-2 CRH family. In Section 5.3, we show how this question (and a related question on statistically hiding commitments) relates to a long-standing question on constructing degree-2 randomized encodings.

5.1 Universal One-way Hash Function

Construction 5.1. Let λ be the security parameter, and let $n = n(\lambda)$, $m = m(\lambda)$, $p = p(\lambda)$ (with $m < n$) be the MQ parameters. Let the MQ distribution $\mathcal{D} = \mathcal{D}_{n,m,p}$ be the uniform distribution that outputs a set of m uniformly random upper-triangular matrices $\mathbf{Q}_i$ and m vectors $\mathbf{L}_i$.

Define the family of hash functions $\mathcal{H}_{n,m,p} = \{h_{\vec{\mathbf{Q}},\vec{\mathbf{L}}} : \mathbb{Z}_p^n \to \mathbb{Z}_p^m \ : \ \vec{\mathbf{Q}} \in (\mathbb{Z}_p^{n\times n})^m,\ \vec{\mathbf{L}} \in (\mathbb{Z}_p^n)^m\}$ as follows:

$$h_{\vec{\mathbf{Q}},\vec{\mathbf{L}}}(\mathbf{x}) = \left(\mathbf{x}^T\mathbf{Q}_i\mathbf{x} + \mathbf{L}_i^T\mathbf{x}\right)_{i=1}^m$$

We now show that under the MQ assumption, this family is universal one-way.

Theorem 5.2. *Let $n = n(\lambda)$, $m = m(\lambda)$, $p = p(\lambda)$ and $\mathcal{D}_{n,m,p}$ be such that (a) $m < n$ and (b) $\mathcal{D}_{n,m,p}$ is the uniform distribution. Then, under the $\mathcal{D}_{n,m,p}$-MQ assumption, the family $\mathcal{H}_{n,m,p}$ is universal one-way.*

The construction and proof immediately generalize to other families of distributions where the distribution of $\mathbf{Q}_i$ is arbitrary but $\mathbf{L}_i$ is still uniformly random. In this exposition, we choose to present the simpler version where both $\mathbf{Q}_i$ and $\mathbf{L}_i$ are uniformly random.

Proof. First, since $m < n$, $\mathcal{H}_{n,m,p}$ is a compressing family of functions.

We now show that it is universally one-way. Assume that there is a PPT UOWHF-breaker algorithm $(\mathsf{Adv}_1, \mathsf{Adv}_2)$. We will construct an algorithm $\mathcal{B}$ that breaks the $\mathcal{D}$-MQ assumption. $\mathcal{B}$ gets as input an MQ challenge $(\mathbf{Q}_i, \mathbf{L}_i)_{i=1}^m$ and does the following.

- Run Adv_1 to get a target input $\mathbf{x}$ and state information r.
- Define the UOWHF family using the matrices $\mathbf{Q}_i^*$ and $\mathbf{L}_i^*$ where:

$$\mathbf{Q}_i^* = \mathbf{Q}_i \text{ and } \mathbf{L}_i^* = \mathbf{L}_i - (\mathbf{Q}_i^* + (\mathbf{Q}_i^*)^T)\mathbf{x} \tag{5.1}$$

- Feed $\vec{\mathbf{Q}} := (\mathbf{Q}_1^*, \ldots, \mathbf{Q}_m^*)$ and $\vec{\mathbf{L}} := (\mathbf{L}_1^*, \ldots, \mathbf{L}_m^*)$ to Adv_2 together with the state information r. Get back a colliding input $\mathbf{y}$, output $\mathbf{y} - \mathbf{x}$ and halt.

First note that the distribution of $\mathbf{Q}_i$ and $\mathbf{L}_i$ from equation 5.1 are uniformly random and hence, the adversary $\mathsf{Adv} = (\mathsf{Adv}_1, \mathsf{Adv}_2)$ will find a colliding input $\mathbf{y}$ with non-negligible probability $1/q(\lambda)$.

Let $\Delta := \mathbf{y} - \mathbf{x}$. Since $\mathbf{x}$ and $\mathbf{y}$ are colliding inputs, we have

$$(\mathbf{x} + \Delta)^T \mathbf{Q}_i^* (\mathbf{x} + \Delta) + (\mathbf{L}_i^*)^T (\mathbf{x} + \Delta) = \mathbf{x}^T \mathbf{Q}_i^* \mathbf{x} + (\mathbf{L}_i^*)^T \mathbf{x}$$

for all $i \in [m]$. A quick calculation then tells us that

$$\Delta^T \mathbf{Q}_i \Delta + \Delta^T \mathbf{Q}_i \mathbf{x} + \mathbf{x}^T \mathbf{Q}_i \Delta + (\mathbf{L}_i^*)^T \Delta = \Delta^T \mathbf{Q}_i \Delta + \left(\mathbf{x}^T \mathbf{Q}_i^T + \mathbf{x}^T \mathbf{Q}_i + (\mathbf{L}_i^*)^T \right) \Delta = 0$$

which in turn gives us

$$\Delta^T \mathbf{Q}_i \Delta + \mathbf{L}_i^T \Delta = 0 \,,$$

by our definition of $\mathbf{L}_i := \mathbf{L}_i^* + (\mathbf{Q}_i^* + (\mathbf{Q}^*)^T)\mathbf{x}$. Thus, $\mathcal{B}$ outputs a solution to the challenge MQ instance with probability $1/p(\lambda)$ as well, which shows that $\mathcal{H}_{n,m,p}$ is a universal one-way hash family. ∎

*Generalizing to Other MQ Distributions. Our proof readily generalizes to distributions for the MQ problem which output an arbitrary distribution of the quadratic forms $\mathbf{Q}_i$ and a uniformly random distribution of the linear forms $\mathbf{L}_i$.

*Generalizing to Larger Degrees. Our construction and proof also generalize to the setting where the hash function has degree d and can be based on the hardness of solving random degree-d polynomial equations, a generalization of the MQ assumption. The reason for considering this generalization is to achieve better shrinkage. The MQ construction can shrink n bits to no less than $m = \sqrt{n}$ bits, since the MQ assumption is false for smaller m. Since we do not know attacks against the degree-d assumption with shrinkage $m = n^{\Omega(1/d)}$, this variant will give us larger shrinkage at the expense of larger degree (and a different assumption).

5.2 Finding Collisions in Random Degree-2 Functions

We now show that the hash function family $\mathcal{H}_{n,m,p}$ is *not* collision-resistant. We remind the reader that $\mathcal{H}_{n,m,p}$ refers to the function where $\mathbf{Q}_i$ are uniformly random upper-triangular matrices and $\mathbf{L}_i$ are uniformly random vectors. The attack we describe below was discovered by Ding and Yang [30] but without a proof of correctness.

Theorem 5.3. *For every $n, m < n$ and p, there is a* $\mathsf{poly}(n, m, \log p)$*-time algorithm* ColFinder *such that*

$$\Pr_{h \leftarrow \mathcal{H}_{n,m,p}}[\mathsf{ColFinder}(h) = (\mathbf{x}, \mathbf{y}) : \ h(\mathbf{x}) = h(\mathbf{y}) \wedge \mathbf{x} \neq \mathbf{y}] = \Omega(1)$$

In other words, the family $\mathcal{H}_{n,m,p}$ is not collision-resistant.

Proof. The strategy of ColFinder is simple. It chooses a uniformly random $\Delta \in \mathbb{Z}_p^n$ and solves the system of equations

$$\left\{ (\mathbf{x}+\Delta)^T \mathbf{Q}_i (\mathbf{x}+\Delta) + \mathbf{L}_i^T (\mathbf{x}+\Delta) = \mathbf{x}^T \mathbf{Q}_i \mathbf{x} + \mathbf{L}_i^T \mathbf{x} \right\}_{i \in [m]}$$

This is in fact a linear system of equations in the unknown $\mathbf{x}$:

$$\left\{ \mathcal{L}_i : \mathbf{x}^T (\mathbf{Q}_i^T + \mathbf{Q}_i)\Delta + \mathbf{L}_i^T \Delta = 0 \right\}_{i \in [m]}$$

where all computations are done mod p. Any solution $\mathbf{x}$ to this system of linear equations gives us a collision $(\mathbf{x}, \mathbf{x}+\Delta)$. Conversely, if there exists a collision with difference Δ, ColFinder will find such a collision.

It remains to show that for uniformly random upper-triangular matrices $\mathbf{Q}_i$ (and possibly uniformly random Δ and $\mathbf{L}_i$), this system has a solution with high probability. We show a stronger statement: namely, that the n-by-m matrix $\widetilde{\mathbf{Q}}$ defined as

$$\widetilde{\mathbf{Q}} = \begin{bmatrix} | & \vdots & | \\ (\mathbf{Q}_1^T + \mathbf{Q}_1)\Delta & \vdots & (\mathbf{Q}_m^T + \mathbf{Q}_m)\Delta \\ | & \vdots & | \end{bmatrix}$$

has full rank, namely rank m a constant probability. This in turn implies that the equations $\mathcal{L}_i$ are guaranteed to have a solution with constant probability.

Let us now bound the probability that $\widetilde{\mathbf{Q}}$ has rank less than m. To do so, let us understand the distribution of $\widetilde{\mathbf{Q}}$ using the following observations.

- Fix an $i \in [m]$ be such that $\Delta_i \neq 0$. Such an i is guaranteed to exist as $\Delta \neq \mathbf{0}$.

  We now claim that for each $j \in [m]$, the entries of $(\mathbf{Q}_j + \mathbf{Q}_j^T)\Delta$ are uniformly random except possibly for the i-th entry. This follows from the fact that the i-th column of $(\mathbf{Q}_j + \mathbf{Q}_j^T)$ is uniformly random (except for its i-th entry) and uncorrelated to the rest of the matrix except the i-th row (which is identical to the i-th column by symmetry).

- This implies, in turn, that $\widetilde{\mathbf{Q}}$ has a uniformly random $(n-1)$-by-m submatrix. The probability that this is full-rank, namely rank $\mathsf{min}(n-1, m) = m$, is a constant (see, e.g., [17]).

Put together, we see that ColFinder succeeds with constant probability in finding a collision. We remark that this argument did not use the randomness of $\mathbf{L}_i$ or Δ, but rather only the fact that the $\mathbf{Q}_i$ are uniformly random upper-triangular matrices. ∎

5.3 Degree-2 CRH via Randomized Encoding?

An intriguing question left open by our work is the existence of a degree-2 CRH. The same question is open also for the related primitive of non-interactive statistically hiding commitments (SHC), which can be easily constructed from a CRH (see Appendix A). We relate these questions to questions about the existence of degree-2 statistically private randomized encodings that were left open by [51].

We start by defining a relaxation of the perfect notion of randomized encoding from Definition 2.4 that allows for statistical privacy error and eliminates the balanced simulation and length requirements.

Definition 5.4 (Statistically-Private, Perfectly Correct Randomized Encoding)**.** Let $f : \{0,1\}^n \to \{0,1\}^m$ be a function. We say that a function $\hat{f} : \{0,1\}^n \times \{0,1\}^\rho \to \{0,1\}^s$ is an *ϵ-private, perfectly correct randomized encoding* (ϵ-RE) of f if there exists a deterministic decoding algorithm C and a randomized simulator S which satisfy the following:

- (Perfect correctness.) For every input $x \in \{0,1\}^n$ and $r \in \{0,1\}^\rho$, it holds that $C(\hat{f}(x;r)) = f(x)$.
- (ϵ-privacy) For every $x \in \{0,1\}^n$, the distribution $\hat{f}(x;r)$, induced by a uniform choice of $r \xleftarrow{R} \{0,1\}^\rho$, satisfies $\mathsf{SD}(\hat{f}(x;r), S(f(x))) \leq \epsilon$, where SD denotes statistical distance.

For the dual notion of ϵ-correct, perfectly-private RE, it is shown in [51] that only very special functions admit such an encoding with a degree-2 $\hat{f}$ (in the Boolean case, this class of functions includes only degree-2 polynomials in the input and functions that test whether the input is in an affine subspace of $\{0,1\}^n$). It is open whether the same holds for the above notion of ϵ-RE.

Question 5.5. Does every finite $f : \{0,1\}^n \to \{0,1\}$ admit a family of degree-2, ϵ_t-private, perfectly correct randomized encodings $\hat{f}_t : \{0,1\}^n \times \{0,1\}^{\rho_t} \to \{0,1\}^{s_t}$ with $\epsilon_t = \mathrm{neg}(t)$ and $\rho_t, s_t = \mathrm{poly}(t)$?

This question is open even for the specific function $f : \{0,1\}^4 \to \{0,1\}$ defined by $f(a,b,c,d) = abc \oplus d$ and even with fixed ϵ (say $\epsilon = 1/3$). In fact, it follows from [1] that the latter function is complete in the sense that an affirmative answer to Question 5.5 for this function implies an affirmative answer for all functions.

We show that an affirmative answer to Question 5.5 together with standard cryptographic assumptions would imply the existence of a degree-2 SHC. Thus, ruling out such an SHC would effectively require settling Question 5.5 in the negative.

A (non-interactive) SHC is defined by a collection $C_z(x,\sigma)$, which given a public random string z (that can be reused for many commitments) maps an input x and secret randomness σ to a commitment c. The commitment c should statistically hide x. It should also be *computationally binding* in the sense that given z it is infeasible to find a pair (x,σ) and (x',σ') which are consistent with the same c, where $x \neq x'$. See Appendix A for a formal definition.

Theorem 5.6. *Suppose there is a CRH or SHC in* NC^1*. Moreover, suppose that the answer to Question 5.5 is affirmative. Then there is a degree-2 SHC.*

Proof. Using Theorem A.2, a CRH in NC^1 implies an SHC in NC^1, which in turn implies an SHC $C_z(x,\sigma)$ in NC^0 [1]. Since every output bit of $C_z(x,\sigma)$ depends on a constant number of input bits, we can apply the degree-2 encoding implied by an affirmative answer to Question 5.5, independently to every output bit of C_z and with $t = |x|$, viewing it as a deterministic function of (x,σ).

This yields a polynomial-size degree-2 function $\hat{C}_z(x, \sigma; \tau)$ which is an ϵ-RE of C_z with ϵ negligible in $k = |x|$. We view $\hat{C}_z$ as a commitment scheme with input x and secret randomness (σ, τ). The perfect correctness requirement of the ϵ-RE implies that $\hat{C}_z$ is computationally binding (since if $(x, (\sigma, \tau))$ and $(x', (\sigma', \tau'))$ violate the binding of $\hat{C}_z$ then (x, σ) and (x', σ') violate the binding of C_z). The statistical hiding property is implied by the fact that the error ϵ of the ϵ-RE is negligible in the input length. Indeed, for every $x \neq x'$ we have

$$\hat{C}_z(x; \sigma, \tau) \approx S_z(C_z(x; \sigma)) \approx S_z(C_z(x'; \sigma)) \approx \hat{C}_z(x'; \sigma, \tau),$$

where σ, τ are uniformly distributed, $\approx$ denotes statistical indistinguishability, and S_z is the simulator of the encoding. ■

For the case of CRH, even an affirmative answer to Question 5.5 does not seem to suffice for a degree-2 construction, since an ϵ-RE of a CRH may lose the shrinking property. Instead, we formulate an ad-hoc variant of randomized encoding that captures a minimal set of requirements needed for respecting the CRH properties.

Definition 5.7 (CRH-Respecting Randomized Encoding)**.** Let $f : \{0,1\}^n \to \{0,1\}^m$ be a function with $m < n$. We say that a function $\hat{f} : \{0,1\}^n \times \{0,1\}^\rho \to \{0,1\}^s$ is a *CRH-respecting randomized encoding* of f if the following hold:

- (Perfect correctness.) There exists a deterministic decoding algorithm C such that for every input $x \in \{0,1\}^n$ and $r \in \{0,1\}^\rho$, it holds that $C(\hat{f}(x; r)) = f(x)$.
- (Injective randomness) For any fixed x, the function $\hat{f}(x; \cdot)$ is injective; namely for any $r \neq r'$ we have $\hat{f}(x; r) \neq \hat{f}(x; r')$.
- (Shrinkage) The encoding $\hat{f}(x; r)$ is shrinking, namely $s < n + \rho$.

Note that given the perfect correctness and injective randomness requirements, the best one can hope for is to match the shrinkage of f, namely the shortest possible encoding output is of size $s = \rho + m$. If the shrinkage requirement of Definition 5.7 is strengthened to require this optimal shrinkage, the definition becomes equivalent to the notion of PRE from Definition 2.4 [1]. In particular, perfect privacy is implied by the perfect correctness, injective randomness, and optimal shrinkage.

On the other hand, even the relaxed requirements of Definition 5.7 *do* imply some weak form of "average-case privacy." Indeed, if the input x could always be recovered from $\hat{f}(x, r)$, then injective randomness guarantees that r can also be recovered, and since $s < n + \rho$ we get a contradiction. Thus, the notion of CRH-respecting randomized encoding crudely has the same flavor of perfect correctness and partial privacy as the notion of ϵ-RE from Definition 5.4.

We now show that applying a CRH-respecting randomized encoding to a CRH indeed yields a CRH. (We use concrete function notation instead of infinite collections of functions for simplicity.)

Claim 5.8. *Suppose $h_z : \{0,1\}^n \to \{0,1\}^m$ is a CRH and $\hat{h}_z : \{0,1\}^n \times \{0,1\}^\rho \to \{0,1\}^s$ is an efficient, CRH-respecting randomized encoding of h_z. Then the function $\hat{h}_z$, viewed as a mapping from $n + \rho$ input bits to s output bits, is a CRH.*

Proof. First, the shrinkage requirement directly guarantees that $\hat{h}$ is shrinking its input (x, r). We show that $\hat{h}$ inherits the collision resistance of h. Suppose that $\mathsf{Adv}(z)$, given a random z, finds a collision $(x, r), (x, r')$ for $\hat{h}_z$, where $(x, r) \neq (x', r')$. By the injective randomness requirement we must have $x \neq x'$ and by perfect correctness we must have $h(x) = h(x')$. Hence, Adv can be used to find a collision (x, x') for h with the same success probability. ■

Finally, we pose a concrete question about the existence of CRH-respecting randomized encodings which is related to the existence of degree-2 CRH.

Question 5.9. Does every $f : \{0,1\}^n \to \{0,1\}^{\lfloor n/10 \rfloor}$ in NC^0 admit (an efficiently computable) degree-2 CRH-respecting randomized encoding?

Under the assumptions of Theorem 4.10, ruling out a degree-2 CRH would require proving a negative answer to Question 5.9.

6 Win-Win Results

In this section we prove that a failure of our constructions leads to interesting algorithmic consequences. Both of our results are based on the following observation.

Lemma 6.1. *Suppose that there exists an algorithm* Adv *with complexity T for which*

$$\Pr_{\mathbf{M} \overset{R}{\leftarrow} \mathbb{Z}_2^{m \times n}} [\mathsf{Adv}(\mathbf{M}) = \mathbf{x} \text{ such that } \mathbf{x} \neq \mathbf{0}, \mathbf{M}\mathbf{x} = \mathbf{0} \text{ and } \Delta(\mathbf{x}) \leq \delta] \geq \epsilon.$$

Then there exists an algorithm Adv′ *with complexity $T' = T + \mathrm{poly}(m)$ such that for every $\mathbf{y} \in \mathbb{Z}_2^m$ it holds*

$$\Pr_{\mathbf{M} \overset{R}{\leftarrow} \mathbb{Z}_2^{m \times n}} [\mathsf{Adv}'(\mathbf{M}, \mathbf{y}) = \mathbf{x} \text{ such that } \mathbf{M}\mathbf{x} = \mathbf{y} \text{ and } \Delta(\mathbf{x}) \leq \delta] \geq \epsilon/2.$$

That is, an algorithm Adv that finds a small subset of the columns of $\mathbf{M}$ that spans the all-zero vector, can be transformed into an algorithm Adv′ that finds a small subset of the columns that spans *any* given target vector $\mathbf{y}$.

Proof. The algorithm Adv′ is given a random matrix $\mathbf{M} \overset{R}{\leftarrow} \mathbb{Z}_2^{m \times n}$ and an arbitrary target vector $\mathbf{y} \in \mathbb{Z}_2^m$. It samples a vector $\mathbf{u} \overset{R}{\leftarrow} \mathbb{Z}_2^n$ and calls Adv with the input $\mathbf{M}' = \mathbf{M} + \mathbf{y} \cdot \mathbf{u}^T$ (note that this is an outer-product). Note that if the output $\mathbf{x}$ of Adv is (1) a valid solution (i.e., it is a non-zero vector of weight at most δn and is in the Kernel of $\mathbf{M}$); and (2) is non-orthogonal to $\mathbf{u}$ (i.e., $\langle \mathbf{x}, \mathbf{u} \rangle = 1$), then $\mathbf{M}\mathbf{x} = \mathbf{M}\mathbf{x} + (\mathbf{y} \cdot \mathbf{u}^T)\mathbf{x} = \mathbf{y}$ and so Adv′ succeeds.

To analyze the success probability, note that the joint distribution of $(\mathbf{M}, \mathbf{u})$ is uniform and therefore (1) happens with probability ϵ, by assumption. Moreover, conditioned on (1), the probability of (2) is exactly $\frac{1}{2}$. The lemma follows. ∎

6.1 bSVP and Distributed Parity-Learning

In this section we show that if the bSVP assumption (as phrased in Corollary 4.7) does not hold, one can learn parities in a distributed setting with non-trivial memory/communication tradeoffs. The influence of memory and communication restrictions on learning in distributed environments has been studied recently by several works (cf. [6, 68, 70, 67] and references therein). We consider here another variant of this question.

Let us first recall the standard notion of PAC-learning parity functions over uniformly sampled examples. For a secret parity function f_s (s is an m-bit vector), the learner is given random (non-noisy) labeled samples of the form (r_i, b_i) where $r_i \overset{R}{\leftarrow} \mathbb{Z}_2^m$ is a random example and $b_i = f_s(r_i) = \langle r_i, s \rangle$ is a binary label. The goal is to predict f_s on a random vector $r^* \overset{R}{\leftarrow} \mathbb{Z}_2^m$ with probability, say, $2/3$. We consider the case where the learner is composed of two parties: a measurement device W that collects the samples and has only limited memory of S bits and no computational power, and

an analyst A who runs in polynomial-time and, given r^*, attempts to predict $f_s(r^*)$. We assume that A sees the vectors r_i's but can access a bit b_i only by reading it from W's memory.[6] Our goal is to minimize the number of bit probes that A makes (subject to the given memory bound S).

In the following we think of a memory-bound of $S = cm$ for some $c > 1$. In this case, W can store cm labels $(b_1, \ldots, b_S)$ and A can trivially achieve a communication of m bits by finding an m-size subset $R' \subset R \stackrel{\Delta}{=} \{r_1, \ldots, r_m\}$ of linearly-independent examples and ask for the corresponding labels. At this point A can recover s and compute $f_s(r^*)$. In fact, A can reduce the communication to $m/2$: first write r^* as a linear combination $\sum v_i R'_i$ of the vectors in R', then ask only for the b_i for which $v_i \neq 0$, and finally compute $f_s(r^*)$ by $\sum v_i b_i$. It is not hard to show (similarly to Observation 3.2) that the expected communication is $m/2$. More generally, the analyst A can achieve a communication of w bits if she can write the challenge vector r^* as a w-weight linear combination of the example vectors R. As shown in Lemma 6.1, this problem reduces to the bSVP problem. In particular, we prove the following lemma.

Lemma 6.2. *Given an efficient algorithm B that solves the $(\alpha, \delta)-$bSVP problem with probability $2/3$ there exists an efficient algorithm that solves the distributed parity-learning problem with memory limitation of m/α and with $m(\delta/\alpha)$ bit probes.*

Proof. The analyst A is given a random matrix of examples $R \stackrel{R}{\leftarrow} \mathbb{Z}_2^{m \times S}$, where $S = m/\alpha$, and a random challenge $r^* \stackrel{R}{\leftarrow} \mathbb{Z}_2^m$. It calls the algorithm B' promised in Lemma 6.1 with input R and target vector r^*. If the algorithm succeeds (which happens with probability $1/3$), the analyst gets a vector v of weight δS for which $Rv = r^*$, and can predict $f_s(r^*)$ using $\delta S = m(\delta/\alpha)$ bit probes. Otherwise, the analysis outputs a random bit. The overall success probability is $\geq 1/3 + 2/3 \cdot (1/2) = 2/3$, as required. ∎

Recall that in Corollary 4.7 we showed that if there exist constants $\delta \in (0, 1/4), \alpha \in (0, 1)$ with $\delta > 2H_2^{-1}(\alpha)$ for which $(\alpha, \delta) -$ bSVP holds, then degree-3 linearly-shrinking CRH exist. Using standard amplification techniques, one can prove a similar result even under the assumption that $(\alpha, \delta) -$ bSVP cannot be solved in polynomial time with probability better than $2/3$.[7] Overall, by combining this with Lemma 6.2 and the approximation of the inverse entropy function from Eq. (2.2), we conclude the following "win-win" result.

Corollary 6.3. *At least one of the following holds:*

- *There exists a degree-3 linearly-shrinking CRH.*
- *For any constant $c > 1$ and any $\gamma > (2/\log c)$, the distributed parity problem can be solved efficiently with memory-bound of cm and γm bit probes for infinitely many m.*

Note that the bit probe rate γ tends to zero when c grows – we are not aware of any efficient solution which achieves such a dependency.

[6] This captures a scenario where the inputs r_i are public (e.g., generated by some environment) but only the measurement device gets to see how the function reacts to it and measure $f_s(r_i)$. We mention that the results of this section carry over (with minor adaptations) to a setting where the r_i's are only given to W.

[7] Indeed, by plugging the weaker assumption in our construction, we get a linearly-shrinking degree-3 "weak"-CRH in which the probability of finding collisions (as defined in Eq. (2.3)) is at most $2/3$ as opposed to negligible. Such a CRH can be amplified into standard CRH while preserving the degree and the linear-shrinkage by expanding the k-bit input x into an $O(k)$-bit vector y via a linear (fixed) error correcting code and then shrinking y down to $(1-\epsilon)k$-long vector, by applying independent copies of the weak CRH to distinct blocks of size $\sqrt{k}$, see [25] for further details. (In fact, one can even get a locality-preserving transformation [3, Lemma 5.7].)

6.2 Speeding-up the BKW algorithm

We move on to the more traditional setting of *learning parity with noise* (LPN) [18]. Recall that in this setting the learner is given random noisy labeled samples of the form (r_i, b_i) where $r_i \stackrel{R}{\leftarrow} \mathbb{Z}_2^m$ is a random example and $b_i = f_s(r_i) + \chi_i$ is a binary label where f_s is a parity function (specified by a secret $s \in \{0,1\}^m$) and χ_i is a random variable which takes the value 1 with probability τ for some noise parameter $\tau \in (0, \frac{1}{2})$. The learner should be able to recover s with, say, probability $2/3$ while minimizing the running time and the sample complexity.[8]

The best known algorithm for solving the LPN problem (i.e., to recover s), due to Blum, Kalai and Wasserman [19], runs in time (and sample) complexity of $2^{O(m/\log m)}$. We show that either the complexity can be reduced to $2^{cm/\log m}$ for arbitrary small constant $c > 0$, or linearly-shrinking CRH of logarithmic degree exist. To this end, we consider the bSVP assumption (for random linear codes) in the polynomial regime, i.e., when the number of columns n is polynomially larger than the number of rows m.

Assumption 6.4. There exists a positive constant $a > 1$ for which (α, δ)-bSVP holds for $\alpha = 1/n^{1/a}$ and $\delta = 8\alpha/(\log(1/\alpha))$.

The constant 8 in the above is somewhat arbitrary and any constant larger than 2 suffices. It will be useful to state the above assumption in terms of the parameter m (and not n as usual). That is, the assumption asserts that, given a random $m \times (n = m^a)$ binary matrix $\mathbf{M}$, it is hard to find a vector of weight $w = \delta n = \frac{8m}{(a-1)\log m}$ in the kernel of M. Note that, unlike in the previous sections, now the dimensions of the matrix are polynomially related (and not linear) and correspondingly we can ask for a kernel vector of sub-linear weight.

Lemma 6.5. *Suppose that Assumption 6.4 does not hold. Then, for every constant $c > 0$ and constant noise rate $\tau \in (0, \frac{1}{2})$ there exists an algorithm that for infinitely many m's, solves the m-dimensional LPN problem with noise rate τ (i.e., recovers the secret s) in time (and sample complexity) of $N = poly(m) \cdot 2^{cm/\log m}$.*

Proof. We describe an algorithm that in time N computes s_1, the first bit of s, with probability $2/3$. Using standard amplification, and by exploiting the symmetry of the LPN problem, such an algorithm can be converted to an algorithm that recovers s with, say, probability $2/3$ at the expense of increasing the time and sample complexity by a factor of $q(m)$ for some fixed (universal) polynomial $q(\cdot)$. (First reduce the error below $1/3m$ via repetition and majority vote, and then apply the amplified algorithm m times where in each iteration we recover the i-bit of s by rotating the examples $i-1$ coordinates to the left.)

Let $a > 1$ be a constant whose value will be determined later, and let $w = \frac{8m}{(a-1)\log m}$. Since Assumption 6.4 does not hold, there exists, by Lemma 6.1, a polynomial-time algorithm Adv_a such that for infinitely many m's, for all $\mathbf{y} \in \mathbb{Z}_2^m$

$$\Pr_{\mathbf{M} \stackrel{R}{\leftarrow} \mathbb{Z}_2^{m \times m^a}} [\mathsf{Adv}_a(\mathbf{M}, \mathbf{y}) = \mathbf{x} \text{ such that } \mathbf{Mx} = \mathbf{y} \text{ and } \mathbf{x} \text{ has Hamming weight of at most } w],$$

is larger than some inverse polynomial $\epsilon(m)$.

Let $n = m^a$, $\mu = 1 - 2\tau$, $t = O(\mu^{2w})$ and $t' = O(t/\epsilon)$. The algorithm asks for nt' labeled examples and partitions them into t' sets T_i each of n examples. For each set T_i, let S_i denote the set of examples without their noisy labels. We apply Adv_a to S_i and set the target vector $\mathbf{y}$ to be

[8] Again, we could define the goal as predicting the value $f_s(r^*)$ for random r^* with non-trivial success probability. However, in this setting the two goals reduces to each other with polynomial overhead.

the first unit vector $\mathbf{e}_1$. If Adv_a succeeds (i.e., returns a subset $S'_i \subseteq S_i$ of size $\frac{bm}{\log m}$ whose sum is $\mathbf{e}_1$) then we XOR together the labels that correspond to the vectors in S'_i and record the result as a vote v_j (which serves as a guess for s_1). If the algorithm fails, we do not record the vote. Finally, we output the majority of all recorded votes v_j.

Analysis: First we claim that each recorded vote is correct (equals to s_1) independently with probability $1/2 + \mu^w$. Indeed, each vote is of the form $s_1 + \sum_{i=1}^{w} \chi_i$ where the χ_i's are independent Bernoulli variables each with expectation τ. It is well known (e.g., [19, Lemma 4]) that $\chi = \sum_{i=1}^{w} \chi_i$ is a Bernoulli random variable with expectation of $(1-\mu^w)/2$ for $\mu = 1 - 2\tau$.

Next, we argue that, except with probability $1/10$, there is a large number of votes. Indeed, each invocation of Adv_a succeeds with at least ϵ probability hence, by Markov's inequality, the probability that there are less than t successful invocations out of, say $t' = 10t/\epsilon$ attempts, is at most $1/10$.

Finally, conditioned on having t votes, by a Chernoff bound, the probability that the final outcome is wrong is $1/10$. The claim follows by a union bound.

Overall, the time complexity of the algorithm is

$$O(nt') = O(m^a \cdot (1-2\tau)^{\frac{20m}{(a-1)\log m}}) = O(m^a \cdot e^{\frac{8m}{\tau(a-1)\log m}}),$$

where e is the natural exponent. Hence, we get the desired running time by taking $a = a(\tau, c)$ to be a sufficiently large constant. ∎

Next we show that if Assumption 6.4 holds then we get CRH of logarithmic degree.

Lemma 6.6. *Under Assumption 6.4, there exists a linearly-shrinking CRH with logarithmic degree.*

Proof. Let $a > 1$ be the constant promised by Assumption 6.4 and let $\alpha = \frac{1}{n^{1/a}}$, $\delta = \frac{8\alpha}{\log(1/\alpha)}$. Also, let

$$d = \log(2/\delta) = \log(n^{1/a}) + \log\log(n^{1/a}) - 2, \quad \beta = 2^{-d},$$

$$b = \frac{2^d}{d} = \frac{n^{1/a}\log(n^{1/a})}{4(\log(n^{1/a}) + \log\log(n^{1/a}) - 2)}.$$

We instantiate Construction 4.1 with the (α, δ)-bSVP sampler and with the (b, β)-expanding mapping promised in the second item of Lemma 4.3. (This part of the lemma holds for non-constant β as well.) Since $\beta \leq \delta/2$ and $b\alpha < 1/4$, we get, by Lemma 4.2, a linearly shrinking CRH with degree of $d = O(\log n)$. The lemma follows. ∎

We conclude the following corollary.

Corollary 6.7. *At least one of the following holds:*

- *There exists a linearly-shrinking CRH with logarithmic degree.*
- *For any constant $c > 0$ and any constant noise rate $\tau \in (0, \frac{1}{2})$ there exists an algorithm that, for infinitely many m, solves the m-dimensional LPN problem with noise rate τ in time and sample complexity of $poly(m) \cdot 2^{cm/\log m}$.*

7 Conclusions and Open Questions

Under plausible intractability assumptions, we establish the existence of low-complexity cryptographic hash functions that compress the input by (at least) a constant factor. In particular, we construct CRH with linear circuit size, constant locality, or algebraic degree 3 over $\mathbb{Z}_2$ under different flavors of the newly introduced binary SVP (bSVP) assumption. We also establish connections with other problems that either support our assumptions or indicate that further progress may be difficult.

While we provide some evidence supporting the validity of the flavors of bSVP we rely on, including a weak connection with the LPN problem, it is left open to obtain a better understanding of the relation between bSVP and LPN or other well studied cryptographic assumptions. It would also be interesting to obtain similar positive results under better or incomparable assumptions, such as the MQ assumption (that we use to construct degree-2 UOWHFs) or the one-wayness of random local functions (used in [3] for constructing local UOWHFs).

Our work leaves open several other natural questions. One such question is the existence of CRH (or even 2-message statistically hiding commitments) with degree 2 or output locality 3. Another is the maximal achievable compression of a degree-d CRH: The bSVP-based security analysis of our construction can only support a constant compression factor, which seems unlikely to be optimal. (In contrast, for *linear-size* CRH we can provide an arbitrary polynomial compression.) A final question is to understand the collision resistance properties of *random* degree-d mappings. While we rule out collision resistance for $d = 2$ with any non-trivial compression, the question is wide open for $d \geq 3$. It would be interesting to study the maximal compression (if any) for which a random degree-d mapping can be a CRH.

Acknowledgement

We thank Zvika Brakerski for participating in an earlier stage of this project. We also thank David Burstein, Simon Litsyn, Ronny Roth, Ohad Shamir, and David Woodruff for helpful discussions.

This work was done in part while the authors were visiting the Simons Institute for the Theory of Computing, supported by the Simons Foundation and by the DIMACS/Simons Collaboration in Cryptography through NSF grant CNS-1523467. The first author was partially supported by the European Union's Horizon 2020 Programme (ERC-StG-2014-2020) under grant agreement no. 639813 ERC-CLC, by an ICRC grant and by the Check Point Institute for Information Security. The second author was partially supported by a Melvin R. Berlin Fellowship in the Cyber Security Research Program. The second and third authors were partially supported by ERC starting grant 259426. The second, third and fourth authors were partially supported by ISF grant 1709/14, BSF grant 2012378, and NSF-BSF grant 2015782. The third author was additionally supported by a DARPA/ARL SAFEWARE award, NSF Frontier Award 1413955, NSF grants 1228984, 1136174, 1118096, and 1065276, and DARPA through the ARL under Contract W911NF-15-C-0205. The fifth author was partially supported by NSF Grants CNS-1350619 and CNS-1414119, Alfred P. Sloan Research Fellowship, Microsoft Faculty Fellowship, the NEC Corporation, a Steven and Renee Finn Career Development Chair from MIT, and DARPA and U.S. Army Research Office under contracts W911NF-15-C-0226. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense, the National Science Foundation, or the U.S. Government.

References

[1] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography in NC0. *SIAM J. Comput.*, 36(4):845–888, 2006.

[2] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. How to garble arithmetic circuits. *SIAM J. Comput.*, 43(2):905–929, 2014.

[3] Benny Applebaum and Yoni Moses. Locally computable UOWHF with linear shrinkage. In *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, pages 486–502, 2013.

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

[5] Jean-Philippe Aumasson and Willi Meier. Analysis of Multivariate Hash Functions. In *Information Security and Cryptology - ICISC 2007, 10th International Conference, Seoul, Korea, November 29-30, 2007, Proceedings*, pages 309–323, 2007.

[6] Maria-Florina Balcan, Avrim Blum, Shai Fine, and Yishay Mansour. Distributed learning, communication complexity and privacy. In *COLT 2012 - The 25th Annual Conference on Learning Theory, June 25-27, 2012, Edinburgh, Scotland*, pages 26.1–26.22, 2012.

[7] Boaz Barak. How to go beyond the black-box simulation barrier. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, pages 106–115, 2001.

[8] Alexander Barg. Complexity issues in coding theory. *Electronic Colloquium on Computational Complexity (ECCC)*, 4(46), 1997.

[9] Alexander Barg and G. David Forney Jr. Random codes: Minimum distances and error exponents. *IEEE Trans. Information Theory*, 48(9):2568–2573, 2002.

[10] Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer. Decoding random binary linear codes in 2 n/20: How 1 + 1 = 0 improves information set decoding. In *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, pages 520–536, 2012.

[11] Mihir Bellare and Phillip Rogaway. Collision-resistant hashing: Towards making uowhfs practical. In *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings*, pages 470–484, 1997.

[12] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 1–10, 1988.

[13] Côme Berbain, Henri Gilbert, and Jacques Patarin. QUAD: A multivariate stream cipher with provable security. *J. Symb. Comput.*, 44(12):1703–1723, 2009.

[14] Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Smaller decoding exponents: Ball-collision decoding. In *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, pages 743–760, 2011.

[15] Olivier Billet, Matthew J. B. Robshaw, and Thomas Peyrin. On Building Hash Functions from Multivariate Quadratic Equations. In *Information Security and Privacy, 12th Australasian Conference, ACISP 2007, Townsville, Australia, July 2-4, 2007, Proceedings*, pages 82–95, 2007.

[16] Nir Bitansky, Ran Canetti, Alessandro Chiesa, Shafi Goldwasser, Huijia Lin, Aviad Rubinstein, and Eran Tromer. The hunting of the SNARK. *IACR Cryptology ePrint Archive*, 2014:580, 2014.

[17] Johannes Blömer, Richard Karp, and Emo Welzl. The rank of sparse random matrices over finite fields. *Random Struct. Algorithms*, 10(4):407–419, 1997.

[18] Avrim Blum, Merrick L. Furst, Michael J. Kearns, and Richard J. Lipton. Cryptographic primitives based on hard learning problems. In *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, pages 278–291, 1993.

[19] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA*, pages 435–440, 2000.

[20] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo random bits. In *23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982*, pages 112–117, 1982.

[21] Elette Boyle, Niv Gilboa, and Yuval Ishai. Breaking the circuit size barrier for secure computation under DDH. In *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I*, pages 509–539, 2016.

[22] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. *SIAM J. Comput.*, 43(2):831–871, 2014.

[23] David Burshtein and Gadi Miller. Bounds on the performance of belief propagation decoding. *IEEE Trans. Information Theory*, 48(1):112–122, 2002.

[24] Chris Calabro. *The Exponential Complexity of Satisfiability Problems*. PhD thesis, University of California, San Diego, 2009.

[25] Ran Canetti, Ronald L. Rivest, Madhu Sudan, Luca Trevisan, Salil P. Vadhan, and Hoeteck Wee. Amplifying collision resistance: A complexity-theoretic treatment. In *Advances in Cryptology - CRYPTO 2007, 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007, Proceedings*, pages 264–283, 2007.

[26] David Chaum, Claude Crépeau, and Ivan Damgård. Multiparty unconditionally secure protocols (extended abstract). In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 11–19, 1988.

[27] Ronald Cramer, Ivan Damgård, and Ueli M. Maurer. General secure multi-party computation from any linear secret-sharing scheme. In *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding*, pages 316–334, 2000.

[28] Ivan Damgård. Collision Free Hash Functions and Public Key Signature Schemes. In *Advances in Cryptology - EUROCRYPT '87, Workshop on the Theory and Application of of Cryptographic Techniques, Amsterdam, The Netherlands, April 13-15, 1987, Proceedings*, pages 203–216, 1987.

[29] Ivan Damgård, Torben P. Pedersen, and Birgit Pfitzmann. On the existence of statistically hiding bit commitment schemes and fail-stop signatures. *J. Cryptology*, 10(3):163–194, 1997.

[30] Jintai Ding and Bo-Yin Yang. Multivariates polynomials for hashing. In *Information Security and Cryptology, Third SKLOIS Conference, Inscrypt 2007, Xining, China, August 31 - September 5, 2007, Revised Selected Papers*, pages 358–371, 2007.

[31] Ilya Dumer. On minimum distance decoding of linear codes. In Ed. G. Kabatianskii, editor, *Fifth Soviet-Swedish intern. workshop Information theory*, pages 50–52, 1991.

[32] Ilya Dumer, Alexey A Kovalev, and Leonid P Pryadko. Distance verification for LDPC codes. *arXiv preprint arXiv:1605.02410*, 2016.

[33] Ilya Dumer, Daniele Micciancio, and Madhu Sudan. Hardness of approximating the minimum distance of a linear code. *IEEE Trans. Information Theory*, 49(1):22–37, 2003.

[34] Uriel Feige, Jeong Han Kim, and Eran Ofek. Witnesses for non-satisfiability of dense random 3cnf formulas. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*, pages 497–508, 2006.

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

[36] Matthieu Finiasz and Nicolas Sendrier. Security bounds for the design of code-based cryptosystems. In *Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings*, pages 88–105, 2009.

[37] Robert G. Gallager. *Low-Density Parity-Check Codes*. 1963.

[38] M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. 1979.

[39] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 169–178, 2009.

[40] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, pages 75–92, 2013.

[41] Oded Goldreich. Candidate One-Way Functions Based on Expander Graphs. In *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation - In Collaboration with Lidor Avigad, Mihir Bellare, Zvika Brakerski, Shafi Goldwasser, Shai Halevi, Tali Kaufman, Leonid Levin, Noam Nisan, Dana Ron, Madhu Sudan, Luca Trevisan, Salil Vadhan, Avi Wigderson, David Zuckerman*, pages 76–87. 2011.

[42] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Collision-free hashing from lattice problems. *Electronic Colloquium on Computational Complexity (ECCC)*, 3(42), 1996.

[43] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986.

[44] Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. Essential coding theory. Draft of a Book, 2015.

[45] Iftach Haitner, Jonathan J. Hoch, Omer Reingold, and Gil Segev. Finding collisions in interactive protocols - tight lower bounds on the round and communication complexities of statistically hiding commitments. *SIAM J. Comput.*, 44(1):193–242, 2015.

[46] Iftach Haitner, Yuval Ishai, Eran Omri, and Ronen Shaltiel. Parallel hashing via list recoverability. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, pages 173–190, 2015.

[47] Shai Halevi and Silvio Micali. Practical and provably-secure commitment schemes from collision-free hashing. In *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, pages 201–215, 1996.

[48] Masanori Hirotomo, Masami Mohri, and Masakatu Morii. A probabilistic computation method for the weight distribution of low-density parity-check codes. In *Proceedings. International Symposium on Information Theory, 2005. ISIT 2005.*, pages 2166–2170, 2005.

[49] Chun-Yuan Hsiao and Leonid Reyzin. Finding collisions on a public road, or do secure hash functions need secret coins? In *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, pages 92–105, 2004.

[50] Xiao-Yu Hu, Marc P. C. Fossorier, and Evangelos Eleftheriou. On the computation of the minimum distance of low-density parity-check codes. In *Proceedings of IEEE International Conference on Communications, ICC 2004, Paris, France, 20-24 June 2004*, pages 767–771, 2004.

[51] Yuval Ishai and Eyal Kushilevitz. Randomizing polynomials: A new representation with applications to round-efficient secure computation. In *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*, pages 294–304, 2000.

[52] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography with constant computational overhead. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 433–442, 2008.

[53] Ahmet B. Keha and Tolga M. Duman. Minimum distance computation of LDPC codes using a branch and cut algorithm. *IEEE Trans. Communications*, 58(4):1072–1079, 2010.

[54] Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing, May 4-6, 1992, Victoria, British Columbia, Canada*, pages 723–732, 1992.

[55] Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced Oil and Vinegar Signature Schemes. In *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, pages 206–222, 1999.

[56] Aviad Kipnis and Adi Shamir. Cryptanalysis of the HFE public key cryptosystem by relinearization. In *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, pages 19–30, 1999.

[57] Simon Litsyn and Vladimir Shevelev. On ensembles of low-density parity-check codes: asymptotic distance distributions. *IEEE Transactions on Information Theory*, 48(4):887–908, 2002.

[58] Vadim Lyubashevsky and Daniele Micciancio. Generalized compact knapsacks are collision resistant. In *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, pages 144–155, 2006.

[59] Tsutomu Matsumoto and Hideki Imai. Public Quadratic Polynominal-Tuples for Efficient Signature-Verification and Message-Encryption. In *Advances in Cryptology - EUROCRYPT '88, Workshop on the Theory and Application of of Cryptographic Techniques, Davos, Switzerland, May 25-27, 1988, Proceedings*, pages 419–453, 1988.

[60] Ralph C. Merkle. A digital signature based on a conventional encryption function. In *Advances in Cryptology - CRYPTO '87, A Conference on the Theory and Applications of Cryptographic Techniques, Santa Barbara, California, USA, August 16-20, 1987, Proceedings*, pages 369–378, 1987.

[61] Ralph C. Merkle. A Certified Digital Signature. In *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, pages 218–238, 1989.

[62] Silvio Micali. Computationally sound proofs. *SIAM Journal on Computing*, 30(4):1253–1298, 2000. Preliminary version in *FOCS'94*.

[63] Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 33–43, 1989.

[64] Jacques Patarin. Cryptoanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt'88. In *Advances in Cryptology - CRYPTO '95, 15th Annual International Cryptology Conference, Santa Barbara, California, USA, August 27-31, 1995, Proceedings*, pages 248–261, 1995.

[65] Chris Peikert and Alon Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, pages 145–166, 2006.

[66] Tal Rabin and Michael Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority (extended abstract). In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 73–85, 1989.

[67] Ran Raz. Fast learning requires good memory: A time-space lower bound for parity learning. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:19, 2016.

[68] Ohad Shamir. Fundamental limits of online and distributed algorithms for statistical learning and estimation. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 163–171, 2014.

[69] Daniel R. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In *Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31 - June 4, 1998, Proceeding*, pages 334–345, 1998.

[70] Jacob Steinhardt, Gregory Valiant, and Stefan Wager. Memory, communication, and statistical queries. In *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, pages 1490–1516, 2016.

[71] Jacques Stern. A method for finding codewords of small weight. In *Coding Theory and Applications, 3rd International Colloquium, Toulon, France, November 2-4, 1988, Proceedings*, pages 106–113, 1988.

[72] Alexander Vardy. The intractability of computing the minimum distance of a code. *IEEE Trans. Information Theory*, 43(6):1757–1766, 1997.

[73] Christopher Wolf. Multivariate quadratic polynomials in public key cryptography. Cryptology ePrint Archive, Report 2005/393, 2005.

[74] Yang Xiao and Kiseon Kim. Searching the minimum distances of LDPC codes. In *Wireless, Mobile and Multimedia Networks (ICWMMN 2008), IET 2nd International Conference on*, pages 211–214, 2008.

[75] Andrew Chi-Chih Yao. Theory and applications of trapdoor functions (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982*, pages 80–91, 1982.

A Statistically Hiding Commitments

In this section we define a non-interactive notion of *statistically hiding commitments* (SHC), which can be viewed as a randomized version of CRH that achieves an input hiding property. We then show that positive results for CRH apply also to the case of SHC.

Similarly to a CRH, an SHC is defined by a public random string z (which can be reused for many commitments). The string z defines a mapping from an input x and secret randomness σ to a commitment string c. The commitment c should statistically hide x. It should also be *computationally binding* in the sense that given z it is infeasible to find a pair (x, σ) and (x', σ') which are consistent with the same c, where $x \neq x'$. We formalize these requirements below.

Definition A.1 (Statistically Hiding Commitment)**.** A collection of functions

$$\mathcal{C} = \left\{ C_z : \{0,1\}^k \times \{0,1\}^{\rho(k)} \to \{0,1\}^{m(k)} \right\}_{z \in \{0,1\}^{s(k)}}$$

is a (non-interactive) *statistically hiding commitment* (SHC) if the following hold:

- (Efficient evaluation and sampling) There exists a pair of efficient algorithms: (a) a *commitment algorithm* C which given $(z \in \{0,1\}^s, x \in \{0,1\}^k, \sigma \in \{0,1\}^\rho)$ outputs $C_z(x, \sigma)$; and (b) a key-sampling algorithm $\mathcal{K}$ which given 1^k samples a index $z \in \{0,1\}^{s(z)}$.

- (Statistical hiding) For every pair of inputs $x, x' \in \{0,1\}^k$ we have

$$\mathsf{SD}((z, C_z(x,\sigma)), (z, C_z(x',\sigma))) \leq \mathrm{neg}(k),$$

  where SD denotes statistical distance, z is picked by $\mathcal{K}(1^k)$, and σ is picked uniformly from $\{0,1\}^{\rho(k)}$.

- (Computational binding) For every probabilistic polynomial-time adversary Adv it holds that

$$\Pr_{z \xleftarrow{R} \mathcal{K}(1^k)}[\mathsf{Adv}(z) = ((x,\sigma),(x',\sigma')) \text{ s.t. } x' \neq x \text{ and } C_z(x,\sigma) = C_z(x',\sigma')] \leq \mathrm{neg}(k).$$

As in the case of CRH, we consider the efficiency of SHC for any fixed public challenge z, namely z defines a function of x and s. This is justified by the fact that the same z can reused.

We now show a simple transformation of a CRH into an SHC using a randomness extractor [29, 47]. The high level idea is to apply a CRH to a secret random input α, and then mask the SHC input with randomness extracted from α. Since the CRH shrinks the input, there is residual entropy in α even when conditioned on the output of the CRH.

Theorem A.2. *Suppose $\mathcal{H} = \left\{h_z : \{0,1\}^k \to \{0,1\}^{m(k)}\right\}_{z \in \{0,1\}^{s(k)}}$ is a CRH, where $m(k) = \lfloor (1-c)k \rfloor$ for some constant $0 < c < 1$. Let $\mathsf{Ext}_k : \{0,1\}^k \times \{0,1\}^d \to \{0,1\}^{\lfloor ck/3 \rfloor}$ be a strong $(ck/2, \epsilon)$ randomness extractor with error $\epsilon(k) = \mathrm{neg}(k)$.*
Then $\mathcal{C} = \left\{C_z : \{0,1\}^{k'} \times \{0,1\}^{k+d} \to \{0,1\}^{m(k)}\right\}_{z \in \{0,1\}^{s(k)}}$, where $k(k')$ is chosen such that $\lfloor ck/3 \rfloor = k'$ and $C_z(x', (\alpha, \beta)) = (h_z(\alpha), \beta, x' \oplus \mathsf{Ext}_k(\alpha, \beta))$ for $\alpha \in \{0,1\}^k$ and $\beta \in \{0,1\}^d$, is an SHC.

Implementing Ext by a random linear function, the SHC obtained in Theorem A.2 has the same algebraic degree as the underlying CRH. Moreover, if Ext is implemented by the linear-size pairwise independent hash function from [52], the SHC additionally maintains the asymptotic circuit size of the CRH. Thus, we get a linear-size (degree-3) SHC under the assumptions of Theorem 4.10.

Unlike the case of CRH, there is no shrinkage requirement for SHC, hence a degree-3 SHC with locality 4 is implied by the existence of any SHC or CRH in NC^1, which is in turn implied by most standard cryptographic assumptions [1]. However, the existence of degree-2 SHC is left open. Using Theorem A.2, a degree-2 SHC would be implied by the existence of a degree-2 CRH, which is one of the main questions left open by this work.