Nearly Optimal Property Preserving Hashing

Justin Holmgren[1], Minghao Liu[2], LaKyah Tyner[2], and Daniel Wichs[1,2⋆]

[1] NTT Research, Sunnyvale, CA 94085, USA
[2] Northeastern University, Boston, MA 02115, USA

Abstract. *Property-preserving hashing (PPH)* consists of a family of compressing hash functions h such that, for any two inputs x, y, we can correctly identify whether some property $P(x, y)$ holds given only the digests $h(x), h(y)$. In a basic PPH, correctness should hold with overwhelming probability over the choice of h when x, y are worst-case values chosen a-priori and independently of h. In an adversarially *robust* PPH (RPPH), correctness must hold even when x, y are chosen adversarially and adaptively depending on h. Here, we study (R)PPH for the property that the *Hamming distance* between x and y is at most t.

The notion of (R)PPH was introduced by Boyle, LaVigne and Vaikuntanathan (ITCS '19), and further studied by Fleischhacker, Simkin (Eurocrypt '21) and Fleischhacker, Larsen, Simkin (Eurocrypt '22). In this work, we obtain improved constructions that are conceptually simpler, have nearly optimal parameters, and rely on more general assumptions than prior works. Our results are:

- We construct information-theoretic non-robust PPH for Hamming distance via syndrome list-decoding of linear error-correcting codes. We provide a lower bound showing that this construction is essentially optimal.
- We make the above construction robust with little additional overhead, by relying on homomorphic collision-resistant hash functions, which can be constructed from either the discrete-logarithm or the short-integer-solution assumptions. The resulting RPPH achieves improved compression compared to prior constructions, and is nearly optimal.
- We also show an alternate construction of RPPH for Hamming distance under the minimal assumption that standard collision-resistant hash functions exist. The compression is slightly worse than our optimized construction using homomorphic collision-resistance, but essentially matches the prior state of the art constructions from specific algebraic assumptions.
- Lastly, we study a new notion of randomized robust PPH (R2P2H) for Hamming distance, which relaxes RPPH by allowing the hashing algorithm itself to be randomized. We give an information-theoretic construction with optimal parameters.

⋆ Research supported by NSF grant CNS-1750795, CNS-2055510 and the Alfred P. Sloan Research Fellowship.

1 Introduction

This work studies the problem of how to compress a large input into a small digest that nevertheless preserves some class of properties of the input. This high level goal is of central importance and lies behind many prominent topics in computer science, such as sketching algorithms, locality sensitive hash functions, streaming algorithms, and compressed sensing.

We focus on an important variant of this problem, called *(robust) property-preserving hashing (R)PPH*, which was recently introduced by Boyle, LaVigne and Vaikuntanthan [BLV19] and further studied by Fleischhacker, Simkin [FS21] and Fleischhacker, Larsen, Simkin [FLS22]. An (R)PPH for a property P (a binary predicate) consists of a compressing family of deterministic hash function h such that, for any x, y, we can determine whether the property $P(x, y)$ holds given only the digests $h(x), h(y)$. In more detail, there is an Eval procedure that operates on the digests and whose goal is to ensure correctness: $\mathsf{Eval}(h(x), h(y)) = P(x, y)$.[3] The basic notion of PPH requires that correctness holds with overwhelming probability over the choice of h, when the inputs x, y are worst-case values chosen *ahead of time and independently of the choice of* h. A robust PPH (RPPH), on the other hand, requires that correctness holds with overwhelming probability over the choice of h, even when the inputs x, y are chosen by an adversary *adaptively depending on the hash function* h. The difference between non-robust and robust PPH is exemplified by the difference between universal hashing and collision-resistant hashing. Concretely, if we consider the *equality property* "$P(x, y) = 1$ iff $x = y$", then universal hashing gives an information-theoretic (non-robust) PPH for equality, while collision-resistant hashing gives an RPPH for equality.

The interesting problem is to construct (R)PPH for more complex properties beyond equality. Most naturally, we'd like to do so for properties $P(x, y)$ that hold if x, y are "similar" in some metric. For example, Apple recently suggested a method for privately detecting users who store known Child Sexual Abuse Material (CSAM) [App, NYT, Scha]. A key component of their system was a hash function called NeuralHash, which was essentially intended to be an RPPH for the property that two images are similar. However, it became clear that NeuralHash is not robust, and it is possible to adversarially find images that are completely different, yet their hashes identify them as being similar [Schb, Cru]. This leads to privacy violations in the overall system, which is one of the reasons that Apple ended up abandoning the CSAM detection system for the time being. The above highlights the need for a better understanding of RPPH, what it can achieve, and what are its limitations.

(R)PPH for Hamming Distance. In this work, following prior works [BLV19, FS21, FLS22], we study (R)PPH for Hamming distance over the binary alphabet. In particular, for some distance bound t, we consider the property $P(x, y)$,

[3] Technically, the Eval procedure also takes as input the description of the hash function h, but for simplicity we omit this throughout the introduction.

which holds iff the Hamming distance between x, y is $||x - y||_0 \leq t$. There are several reasons for focusing on Hamming distance. Firstly, Hamming distance is arguably the most basic metric to study and understanding it is likely to be a prerequisite for understanding more complex metrics. Secondly, a common approach to defining "similarity" between complex objects is to first translate these objects into binary "feature vectors" that represent a list of potential features and indicates whether or not the object has them, and then looking at the Hamming distance between the feature vectors. In this case, a good (R)PPH for Hamming distance gives a good (R)PPH for testing similarity of more complex objects. Lastly, by focusing on Hamming distance, we can make use of an extensive set of tools from coding theory to help us along.

The main measure of efficiency that we seek to optimize is the output length m of the (R)PPH, as a parameter that depends on the input length n, the distance parameter t and the security parameter λ. In particular, we would like the RPPH to be as compressing as possible by minimizing m for any choice of n, t, λ.

Prior Work. The work of Boyle, LaVigne and Vaikuntanthan [BLV19] initiated the general study of (R)PPH. They provided definitions for both the non-robust and robust variants of PPH.[4] Although [BLV19] does not directly offer any constructions of RPPH for the exact Hamming distance property studied here, their main positive results consider a relaxation called RPPH for *gap-Hamming distance*, where the goal is only to distinguish between the case where the Hamming distance between x and y is $\leq t$ versus $> (1 + \delta)t$, for some distance t and gap parameter $\delta > 0$. In other words, this relaxation only requires $\mathsf{Eval}(h(x), h(y))$ to output 1 in the former case and 0 in the latter case, but any output is permissible in the gap between them. The work of [BLV19] gave two constructions of RPPH for gap-Hamming distance with any constant gap $\delta > 0$. The first construction is based on only the existence of collisions-resistant hash functions (CRHFs). Assuming CRHFs with output length $\ell = \ell(\lambda)$, they showed that for any constant *compression factor* $\eta > 0$, there exists some constant $\rho > 0$ such that for any distance $t \leq \rho \cdot n/(\ell \cdot \log \ell)$, there is an RPPH for gap Hamming distance with output length $m \leq \eta \cdot n$.[5] Their second construction is based on a new but plausible computational assumption that they introduce and call the

[4] They also considered two additional intermediate variants of PPH, where the adversary does not get the full description of the hash function but gets some partial oracle access before choosing x, y. Our notion of robust PPH is the strongest notion they considered and is also referred to as a "direct access robust" PPH in their work.

[5] Asymptotically, the existence of CRHFs with output length $\ell(\lambda) = \lambda$ is equivalent to those with output length $\ell(\lambda) = \lambda^{\varepsilon}$ for $\varepsilon > 0$. Moreover, it may be plausible to even conjecture the existence of CRHFs with (e.g.,) output length $\ell(\lambda) = \log \lambda \log \log \lambda$. However, these choices will have vastly different exact security. All the constructions/reductions referred to in this work preserve exact security. Therefore, we find it more informative to phrase all results in terms of the exact output length $\ell(\lambda)$ of the underlying primitive and the construction with inherit the exact security of that primitive with the given output length.

Sparse Short Vector (SSV) assumption. Under that assumption, they got somewhat better parameters, showing that for any constant $\eta > 0$, there exists some constant $\rho > 0$ such that for any $t \leq \rho \cdot n / \log n$ there is an RPPH for gap Hamming distance with output size $m \leq \eta \cdot n$.

The work of Fleischhacker, Simkin [FS21] gave the first construction of RPPH for exact Hamming distance. They did so under a new assumption in bilinear groups, which they called the *q-Strong Bilinear Discrete Logarithm (q-SBDL) Assumption*. They showed that, assuming q-SBDL holds in a group whose elements can be represented using $\ell = \ell(\lambda)$ bits, for any distance t there is an RPPH for exact Hamming distance with output length $m = O(t\ell)$. In particular, the RPPH is non-trivially compressing for $t = O(n/\ell)$.

The work of Fleischhacker, Larsen, Simkin [FLS22], gave a similar result as above but under the *Short-Integer Solution (SIS)* assumption, which is a well-studied assumption (it is implied by learning-with-errors (LWE)) and can be based on the hardness of worst-case lattice problems [Ajt96]. In particular, under the SIS assumption, they showed that for any distance t there is an RPPH for exact Hamming distance with output size $m = O(t\ell \cdot \log n)$, where $\ell = \ell(\lambda)$ is the output length of Ajtai's hash function based on SIS. In particular, the output size is non-trivially compressing for $t = O(n/(\ell \cdot \log n))$.

To summarize, the best prior RPPH constructions for exact Hamming distance at the very least required output size $m \geq t\ell(\lambda)$, where ℓ is some polynomial. They also required specific algebraic assumptions, namely SIS or the q-SBDL assumption in bilinear maps. For gap Hamming distance, we knew how to get slightly better output length $m \geq t \log n + \ell(\lambda)$, but only under a new non-standard variant of SIS, or we knew how to get $m \geq t\ell(\lambda) \log n$ under just collision-resistance.

1.1 Our Results

In this work, we give new constructions of (R)PPH for exact Hamming distance. Our constructions are conceptually simpler, are based on more general assumptions, and achieve improved compression compared to prior work. Our results are as follows.

Non-Robust PPH. Our first result is to construct a *non-robust PPH for Hamming distance* via a simple connection to syndrome list-decoding of linear error-correcting codes. In terms of parameters, the output size of our hash function is $m = \eta \cdot n + \lambda$, where $1 - \eta$ is the optimal rate of a linear list-decodable error-correcting code that can correct t errors. Inefficiently, we can go up to the Hamming bound with $\eta = H(t/n)$, where H is the Shannon binary entropy function. Efficiently we can go up to the slightly weaker *Blokh-Zyablov* bound. In either case, this implies $m = O(t \log n) + \lambda$. However, it also implies non-trivial compression for larger distances up to $t = O(n)$. In particular, for any constant compression factor $\eta > 0$ there exists some $\rho > 0$ such that there is a (non-robust) PPH with output length $m = \eta n + \lambda$ for all distances $t \leq \rho \cdot n$.

We give a matching lower bound, showing that the output size has to satisfy $m > (H(t/n) - o(1)) \cdot n$.

RPPH from Homomorphic Collision-Resistance. Our next result extends the above idea to add robustness and achieve *RPPH for Hamming distance* by leveraging *homomorphic collision-resistant hash functions*, which we in turn construct under either the standard *discrete logarithm (DLOG)* assumption or the *short-integer-solution (SIS)* assumption. The construction adds a constant factor overhead of at most $(\log_2 3)$ compared to our non-robust PPH, giving $m = (\log_2 3)\eta \cdot n + \ell$, where $1 - \eta$ is the optimal rate of a linear list-decodable error-correcting code that can correct t errors, and $\ell = \ell(\lambda)$ is the output length of the homomorphic CRHF (e.g., the bit-length of a group element). In particular, our output length is bounded by $m = O(t \cdot \log n + \ell)$, while previous constructions [FS21, FLS22] achieved $m = O(t \cdot \ell)$. Since we always assume $n = \text{poly}(\lambda)$, we can conclude that $\log n = O(\log \lambda)$ is asymptotically smaller than $\ell = \text{poly}(\lambda)$. Moreover, for any constant compression factor $\eta > 0$ there exists some constant $\rho > 0$ such that we get an RPPH for Hamming distances $t \leq \rho \cdot n$ with output size $m \leq \eta \cdot n$. Previous constructions of RPPH for Hamming distance [FS21, FLS22] only achieved non-trivial compression $\eta < 1$ for sub-linear distances $t = O(n/\ell)$, while we do so for up to linear distances $t = O(n)$.

RPPH from Standard Collision-Resistance. We also construct the first RPPH for Hamming distance based on the minimal assumption that (standard) collision-resistant hash functions (CRHFs) exist. Previously, we only knew how to do this for gap-Hamming distance [BLV19], while we show how to do for exact Hamming distance. In fact, we show how to using syndrome decoding to generically upgrade an RPPH for gap-Hamming distance into one for exact Hamming distance. The achieved parameters are slightly worse than those of our optimized construction based on homomorphic CRHFs above, but are comparable to those achieved by prior constructions for exact Hamming distance [FS21, FLS22] based on specific algebraic assumptions. In particular, assuming CRHFs with output length $\ell = \ell(\lambda)$, we get an RPPH for distance t with output length $m = O(t \cdot \ell \cdot \log(n/t))$.

Randomized RPPH (R2P2H). We also consider a *randomized* notion of RPPH (R2P2H), where the computations of the hash function $h(x)$ can itself be a randomized. The adversary can choose worst-case values x, y after seeing the description of h, but before knowing the internal randomness that will be employed in the computation of $h(x), h(y)$. The adversary wins if $\mathsf{Eval}(h(x), h(y)) \neq P(x, y)$, and we require that the can only happen with negligible probability over the choice of the hash function h and the internal randomness used to compute $h(x), h(y)$. We emphasize that, aside for allowing the hash function to be randomized, the security guarantee provided by R2P2H is also qualitatively weaker than that of deterministic RPPH. For deterministic RPPH, the security definition implicitly allows the adversary to choose y after seeing $h(x)$, since the adversary can compute $h(x)$ himself. This is not the case for R2P2H, where seeing $h(x)$ can

reveal something about the internal randomness employed in the computation that would allow the adversary to find a bad y that breaks security. Surprisingly, this relaxation to R2P2H allows us to get non-trivial information-theoretic constructions. It was previously known that one can achieve information-theoretic R2P2H for the equality predicate, where the output length is $m = O(\sqrt{n})$ and that this is optimal [NS96, BK97, MNS08, CN22]. We extend this to showing a construction of information-theoretic R2P2H for Hamming distance t, where the output length is $O(\sqrt{\lambda n}\log n) + \eta n)$, where $1 - \eta$ is the optimal rate of a linear list-decodable error-correcting code that can correct t errors; in particular $\eta n \leq O(t \log n)$.

1.2 Our Techniques

On a technical level, our constructions are quite different than those of [FS21, FLS22]. The common theme of all our results is the reliance on syndrome decoding.

PPH from Syndrome Decoding. We start with a simple construction of a non-robust PPH based on *syndrome (list) decoding* of linear error-correcting codes. Assume there exists some linear error-correcting code over a field $\mathbb{F}$, having codeword length n, message length k, and the ability to (efficiently) correct up to t errors. This is equivalent to the existence of a *parity check* matrix $P \in \mathbb{F}^{(n-k)\times n}$ such that for any *error-vector* $e \in \mathbb{F}^n$ with Hamming weight $\|e\|_0 \leq t$ we can (efficiently) recover e from the *syndrome* $P \cdot e$. More generally, a code that allows (efficient) list-decoding of up to t errors implies that given a syndrome $P \cdot e$ as above we can (efficiently) recover a polynomial-sized list $\mathcal{L}$ of potential error vectors with the guarantee that $e \in \mathcal{L}$. Without loss of generality, we can assume that each $e_i \in \mathcal{L}$ has Hamming-weight $\|e_i\|_0 \leq t$.

For our construction of PPH, assume we have a list-decodable code as above over the binary field $\mathbb{F}_2$ and let P be the parity check matrix. We will also make use of the universal hash function $h_{univ}(x) = A \cdot x$ where $A \leftarrow \mathbb{F}_2^{\lambda \times n}$ is a random matrix. This ensures that for any $x_1 \neq x_2 \in \mathbb{F}_2^n$ chosen a-priori, we have $\Pr_A[Ax_1 = Ax_2] = 2^{-\lambda}$. We will rely on the fact that this universal hash function is linearly homomorphic with $h_{univ}(x_1) - h_{univ}(x_2) = x_1 - x_2$.[6]

The description of the PPH h consists of the random matrix A of the universal hash function. Given an input x, the PPH output $y = h(x)$ is defined as $y = (P \cdot x, A \cdot x)$, consisting of the syndrome and the universal-hash of x. Given $y_1 = h(x_1), y_2 = h(x_2)$ with $y_1 = (v_1, w_1), y_2 = (v_2, w_2)$ the procedure $\mathsf{Eval}(y_1, y_2)$ does the following. It runs the syndrome list-decoding algorithm on $v_1 - v_2 = P \cdot (x_1 - x_2)$ to recover a list of potential error-vectors $\mathcal{L}$. If there exists some $e_i \in \mathcal{L}$ such that $A \cdot e_i = w_1 - w_2$ than the Eval algorithm accepts else it rejects.

To see that the above construction satisfies the definition of a PPH, we consider two cases. First, suppose x_1 and x_2 are "close": i.e., $\|x_1 - x_2\|_0 \leq t$.

[6] Since we're working over $\mathbb{F}_2$, addition and subtraction are equivalent, but we use subtraction to make it easier to compare to later constructions that work in larger fields.

Then, during the computation of $\mathsf{Eval}\big(h(x_1), h(x_2)\big)$, the correctness of syndrome list-decoding for the syndrome $v_1 - v_2 = P \cdot (x_1 - x_2)$ ensures that $x_1 - x_2$ appears in the list $\mathcal{L}$. Moreover $A \cdot (x_1 - x_2) = w_1 - w_2$ and therefore the Eval algorithm will accept with probability 1. Next, suppose that x_1 and x_2 are instead "far": i.e., $\|x_1 - x_2\|_0 > t$. Then, during the computation of $\mathsf{Eval}\big(h(x_1), h(x_2)\big)$, no matter what list $\mathcal{L}$ is generated, we know that $x_1 - x_2 \notin \mathcal{L}$ since the list only contains vectors with Hamming weight at most t. Furthermore, the list $\mathcal{L}$ is independent of A and is polynomial in size. This ensures that the probability over A that there exists some $e_i \in \mathcal{L}$ such that $A \cdot e_i = w_1 - w_2 = A \cdot (x_1 - x_2)$ is at most $|\mathcal{L}| \cdot 2^{-\lambda} = \mathrm{negl}(\lambda)$.

The output size of the above PPH is $m = (n - k) + \lambda$ bits, where k is determined by the optimal rate of the code that can list-decode up to t errors. It is known that, inefficiently, such linear list-decodable codes exist with rates k/n arbitrarily close to $1 - H(t/n)$ where H is the binary entropy function [GHK10]. The well-known *Hamming bound* states that it is impossible to do better. This gives an inefficient PPH with output length $m \approx H(t/n) \cdot n + O(\lambda)$. For efficiently list-decodable codes, it is a well known open problem to match the Hamming bound. Instead, the best known constructions [GR09] achieve a slightly worse bound called the *Blokh-Zyablov* bound [BZ82] (see Fact 3 for the exact expression). While this bound is somewhat difficult to interpret, we can always bound the output length by at most $m = O(t \log n + \lambda)$. Moreover, for any constant compression factor $\eta > 0$, there is some constant $\rho > 0$ such that we get a PPH for all distances $t \leq \rho \cdot n$ with output length $m \leq \eta \cdot n + \lambda$.

Remark: Weak Robustness and Heuristic RPPH. As a remark, we mention that we can leverage the result of [BLV19], who showed that one can generically upgrade a non-robust PPH into a weak form of "double-oracle access robust" PPH, where security holds even if the adversary is given oracle access to the hash function h and the evaluation procedure Eval but does not get the code of h itself. This transformation only relies on one-way functions and only adds a small $O(\lambda)$ additive overhead. The idea is to encrypt the output of the non-robust PPH using symmetric-key authenticated encryption whose key is stored as part of the hash function (and which can even be made deterministic), and the Eval procedure first decrypts the non-robust PPH digests and then does what the non-robust Eval procedure would do.

Moreover, we can heuristically convert any such "double-oracle access robust" PPH into a fully robust RPPH by obfuscating the code of the hash function h and the Eval procedure, without increasing the output size of the hash at all. Therefore, this gives heuristic evidence that we can robustly match the above parameters of our non-robust PPH without any additional overhead. Our main results show how to almost match the above parameters robustly under standard assumptions.

Lower Bound. We also prove a lower bound on the output length m of any (not necessarily robust or efficiently computable) PPH for Hamming distance, showing that we require $m > \log \binom{n}{t}$ which implies $m \geq (H(t/n) - o(1)) \cdot n$ and

$m \geq t \log(n/t)$. Our lower bound is simpler than the previous lower bound of [FLS22] and, more importantly, for constant error-rate $\rho = t/n$, it gives a tight bound on compression factor $\eta = m/n$, showing $\eta = H(\rho) - o(1)$. Our upper bound shows how to match the lower bound of $\eta \approx H(\rho)$ inefficiently. Efficiently, our upper bound gives a slightly worse η matching the Blokh-Zyablov bound. Closing this gap between our inefficient and efficient constructions boils down to the fundamental coding theoretic problem of improving the rate of efficiently list-decodable linear codes from the Blokh-Zyablov bound to the better Hamming bound.

The idea behind the lower bound is as follows. For a random x, we show that if we can correctly guess the PPH output $y = h(x)$ as well as some value x' that's exactly at distance t from x, then we can recover x. If we select y, x' uniformly at random then our guess is good with probability $\frac{1}{2^m} \cdot \frac{\binom{n}{t}}{2^n}$, but this can then be at most the probability of guessing x, which is $\frac{1}{2^n}$.

RPPH from Homomorphic Collision-Resistance. We take our construction of PPH for Hamming distance and show how to make it robust. At a high level, the idea is exactly the same as before, and we simply replace the homomorphic universal hash function h_{univ} with a homomorphic collision-resistant hash function h_{CR}. We have such hash functions under the discrete-logarithm (DLOG) assumption – namely, the Pedersen hash function $h_{CR}(x_1, \ldots, x_n) = \prod g_i^{x_i}$, where g_i are random group elements in some prime-order cyclic group. We also have such hash functions under the *short-integer-solution (SIS)* problem – namely, Ajtai's hash function $h_{CR}(x) = A \cdot x$, where A is a random compressing matrix over $\mathbb{Z}_q$ and $||x||_\infty$ is small. In both cases, the output size can be bounded by some polynomial $\ell = \ell(\lambda)$.

There is only one catch: the above hash functions are homomorphic over $\mathbb{Z}_q$ for some $q > 2$ rather than over $\mathbb{Z}_2$. In particular, given $w_1 = h_{CR}(x_1), w_2 = h_{CR}(x_2)$ we can compute $h_{CR}(x_1 - x_2)$ where the subtraction is now over $\mathbb{Z}_q$. Since our PPH construction applies the hash function to values $x_1, x_2 \in \{0,1\}^n$ we have $x_1 - x_2 \in \{-1, 0, 1\}^n$ is the same when computed mod q or over the integers. Therefore, to make the overall PPH construction work, we will also need to use a linear (list decodable) code over some field $\mathbb{F}$ of characteristic $p > 2$ so that, given the syndrome $P \cdot (x_1 - x_2)$ computed over $\mathbb{F}$, we can recover a list containing $x_1 - x_2 \in \{-1, 0, 1\}^n$ computed over the integers. For simplicity, we can just use codes over $\mathbb{F}_3$ instead of $\mathbb{F}_2$. With these changes, the construction and proof of security are essentially the same as in the non-robust case, but now we rely on collision-resistance instead of universality to achieve security even when x_1, x_2 are chosen adaptive depending on the description of the hash function h.

The parameters of the resulting RPPH are essentially the same as those of the non-robust PPH, with the only difference that the hash output now contains $n-k$ elements of $\mathbb{F}_3$ rather than bits. This increases the bit-length of the output

by a multiplicative factor of at most $\log_2(3) \approx 1.58$.[7] It is an interesting open problem to get rid of this constant-factor increase by constructing CRHFs that are homomorphic over $\mathbb{Z}_2^n$.[8]

RPPH from Standard Collision-Resistance. Next, we show how to construct RPPH for Hamming distance using just standard collision-resistant hash functions (CRHFs). The output size is larger than that of our optimized construction using homomorphic collision-resistance, but essentially matches the prior state of the art constructions [FS21, FLS22] from specific assumptions.

Our construction relies on two ingredients. The first is an RPPH for *gap Hamming distance*, which was previously constructed from CRHFs by [BLV19]. Namely, assuming a CRHF with output length $\ell = \ell(\lambda)$, they gave a construction of an RPPH for gap Hamming distance with any constant gap $\delta > 0$ and any constant-factor compression, for distances up to $t = O(n/\ell \log \ell)$. We generalize their analysis to showing that for smaller distances t we can get even smaller compression, and in general, for any t, we can make the output as small as $O(t\ell \log(n/t))$. Our second ingredient is a linear error-correcting code over $\mathbb{F}_2$ with a parity check matrix $P \in \mathbb{F}_2^{(n-k)\times n}$ that enables efficient (unique) syndrome decoding from $(1+\delta)t$ errors.

We use syndrome decoding to upgrade an RPPH for gap Hamming distance h_{gap} with some constant gap $\delta > 0$, into an RPPH for exact Hamming distance h_{exact}. We define $h_{exact}(x) = (P \cdot x, h_{gap}(x))$. Given two hashes $h_{exact}(x_1) = (P \cdot x_1, h_{gap}(x_1))$, $h_{exact}(x_2) = (P \cdot x_2, h_{gap}(x_2))$, we can define the Eval_{exact} procedure that tests whether $||x_1 - x_2||_0 \leq t$ as follows. First, it runs $\mathsf{Eval}_{gap}(h_{gap}(x_1), h_{gap}(x_2))$ and if that outputs 0 then we know that $||x_1 - x_2||_0 > t$ and hence output 0. Otherwise, if Eval_{gap} outputs 1 then we know that $||x_1 - x_2||_0 \leq (1+\delta)t$. In this case we apply syndrome decoding on $P \cdot (x_1 - x_2)$ to uniquely recover $(x_1 - x_2)$ and if the Hamming weight is $\leq t$ we outputs 1 else 0.

The end result is an RPPH for exact Hamming distance, where the output length is the sum of $n-k$ and the output length of the RPPH for gap Hamming distance from CRHFs. Using Reed-Solomon codes, the former can be bounded by $O(t \log n)$. Therefore the second term dominates, and we get the same parameters for exact Hamming distance as the previous construction for gap Hamming distance by [BLV19].

[7] On the other hand, it allows us to use codes over $\mathbb{F}_3$ which may have slightly improved rate compared to ones over $\mathbb{F}_2$.

[8] A heuristic construction would be to define the hash function h_{CR} whose description consists of an obfuscated program that has a hard-coded random matrix $A \leftarrow \mathbb{Z}_2^{\lambda \times n}$ and a key k for a pseudorandom permutation $\pi_k : \{0,1\}^\lambda \to \{0,1\}^\lambda$. On input ("hash", x) the program would output $\pi_k(Ax)$, which we would also define as the output of the hash function $h_{CR}(x)$. On input ("homomorphism", y_1, y_2) the program would output $\pi_k(\pi_k^{-1}(y_1) - \pi_k^{-1}(y_2))$, which would allow us to implement the homomorphic operation on the hash outputs.

R2P2H. Finally, we turn to the construction of randomized RPPH (R2P2H). We go back to our initial construction of a (non-robust) PPH for Hamming distance, where $h(x)$ outputs a syndrome of x and a homomorphic universal hash of x. We can think of the universal hash function as a (deterministic, non-rboust) PPH for equality. By taking that construction and replacing the universal hash function with a R2P2H for quality we get our R2P2H for Hamming distance. We just need the R2P2H for equality to satisfy an appropriate homomorphic property, and we show how to adapt known constructions to do so.

Open Question. While our work gives nearly optimal constructions of (R)PPH for exact Hamming distance, it leaves open the question whether one can get significantly better parameters for gap Hamming distance. Recall that, in the non-robust case, we had a lower bound of $m \geq \log \binom{n}{t} \geq t \log(n/t)$ on the output length of a (even non-robust) PPH for exact Hamming distance. As pointed out in [BLV19], using the result of [KOR00], it turns out that one can do much better for gap Hamming distance: for any constant gap $\delta > 0$ there is a non-robust PPH for gap Hamming distance with output length just $O(\lambda)$ independent of n, t. A very interesting open question is whether it is possible to match this with robustness or not. Currently, we don't even have heuristic constructions that would beat the $m \geq t \log(n/t)$ lower bound in the gap setting with robustness. On the other hand, we also currently don't have any techniques for proving any lower bounds on the cost of robustness – all current lower bounds for RPPH also hold for non-robust PPH.

1.3 Other Related Work

Locality sensitive hash functions [IM98] can be thought of as a strengthening of PPH, where we want $h(x_1) = h(x_2)$ to collide iff $P(x_1, x_2)$ holds. In other words, we can think of this as a special case of PPH where Eval just outputs 1 iff the digests are equal. While there is a simple construction of locality sensitive hash functions for gap Hamming distance [IM98], there are strong lower bounds showing that they cannot achieve a negligible correctness error, even in the non-robust setting [MNP07, OWZ11]. In particular, they cannot be robust.

Secure sketches [DORS08] ensure that, given a hash (called a "sketch") $h(x_1)$ and some x_2 within distance t of x_1, we can recover x_1. While the original notion of secure sketches did not require the digest to be compressing, one of the constructions of [DORS08] for Hamming distance is based on syndrome decoding and is compressing. Secure sketches easily yield a relaxation of PPH (resp. RPPH), where we can determine whether $P(x_1, x_2)$ holds given one digest $h(x_1)$ and the other input x_2 in the clear. In particular, we simply append a universal (resp. collision resistant) hash function of x_1 to the output of the sketch; then, given x_2, we first use the sketch to attempt to recover a candidate x_1' for x_1, then check that it matches the universal (resp. collision resistant) hash, and finally check that it is within distance t of x_2. This type of relaxed notion of (R)PPH was also defined by [BLV19], and referred to as a "single-input property" (R)PPH. Given the above, the main novelty of our work and previous works on

RPPH for Hamming distance, is that we need to decide whether $P(x_1, x_2)$ holds given only the two digests $h(x_1), h(x_2)$, without having either of the inputs in the clear.

2 Preliminaries

Notation. When X is a distribution, or a random variable following this distribution, we let $x \leftarrow X$ denote the process of sampling x according to the distribution X. If X is a set, we let $x \leftarrow X$ denote sampling x uniformly at random from X. We use the notation $[k] = \{1, \ldots, k\}$. If $x \in \{0,1\}^k$ and $i \in [k]$ then we let $x[i]$ denote the i'th bit of x. If $s \subseteq [k]$, we let $x[s]$ denote the list of values $x[i]$ for $i \in s$.

Predictability and Entropy. The *predictability* of a random variable X is $\mathbf{Pred}(X) \stackrel{\mathsf{def}}{=} \max_x \Pr[X = x]$. The *min-entropy* of a random variable X is $\mathbf{H}_\infty(X) = -\log(\mathbf{Pred}(X))$. Following Dodis et al. [DORS08], we define the conditional predictability of X given Y as $\mathbf{Pred}(X|Y) \stackrel{\mathsf{def}}{=} \mathbb{E}_{y \leftarrow Y}[\mathbf{Pred}(X|Y = y)]$ and the (average) conditional min-entropy of X given Y as: $\mathbf{H}_\infty(X|Y) = -\log(\mathbf{Pred}(X|Y))$. Note that $\mathbf{Pred}(X|Y)$ is the success probability of the optimal strategy for guessing X given Y.

Lemma 1 ([DORS08]). *For any random variables X, Y, Z where Y is supported over a set of size T we have $\mathbf{H}_\infty(X|Y,Z) \leq \mathbf{H}_\infty(X|Z) - \log T$.*

Universal Hashing. We recall the definition of universal hash function and a simple well-known construction via matrix multiplication.

Definition 1. *A family of hash functions $\mathcal{H} = \{h : \{0,1\}^n \to \{0,1\}^m\}$ is a* universal hash family *if for all $x_1, x_2 \in \{0,1\}^n$ such that $x_1 \neq x_2$, we have:*

$$\Pr[h(x_1) = h(x_2) : h \leftarrow \mathcal{H}] \leq 2^{-m}$$

We will rely on the following simple universal hash function family, which also has the additional feature of being homomorphic over $\mathbb{Z}_2^n$ with $h(x_1) + h(x_2) = h(x_1 + x_2)$.

Lemma 2. *For any n, m, the hash function family $\mathcal{H}$ consisting of hash functions $h_A(x) = A \cdot x$ with $A \in \mathbb{Z}_2^{m \times n}$, is a universal hash family.*

Proof. Let $x_1, x_2 \in \{0,1\}^n$ such that $x_1 \neq x_2$ and let $v = (x_1 - x_2) \neq 0^n$. Denote the bits of $v = (v_1, \ldots, n_n)$. Then there exists some $i \in [n]$ such that $v_i = 1$. Denote the columns of A by $A = [a_1, \ldots, a_n]$ with $a_i \in \mathbb{Z}_2^m$. Then $\Pr_A[Ax_1 = Ax_2] = \Pr_A[Av = 0] = \Pr_A\left[a_i = -\sum_{j \neq i} a_j \cdot v_j\right] = \frac{1}{2^m}$.

2.1 Coding Theory

Definition 2. *An* $[n,k]_q$ code *for* $n,k,q \in \mathbb{Z}^+$ *is an injective linear function* $C : \mathbb{F}_q^k \to \mathbb{F}_q^n$*. We call* k *the* message length *and* n *the* block length *of* C*.*

A codeword *of* C *is any element of the image of* C*.*

Definition 3 (Parity Checks / Syndromes). *A* parity check matrix[9] *for an* $\mathbb{F}$*-linear code* $C : \mathbb{F}^k \to \mathbb{F}^n$ *is a matrix* $P \in \mathbb{F}^{(n-k)\times n}$ *such that* $c \in \mathbb{F}^n$ *is a codeword of* C *if and only if* $P \cdot c = 0$*.*

When C *and* P *are fixed, we call* $P \cdot y$ *the* syndrome *of* y*.*

Definition 4 (Distance). *The* distance *of a code* C *is the minimum Hamming distance between two different codewords of* C*.*

Definition 5 (List-Decoding). *An* $[n,k]_q$ *code* $C : \mathbb{F}_q^k \to \mathbb{F}_q^n$ *is said to be* combinatorially list decodable against t errors *if for any* $y \in \Sigma_k^n$*, there are at most*[10] $\mathrm{poly}(n)$ *codewords of* C *within Hamming distance* t *of any* $y \in \mathbb{F}_q^n$*. If there is a* $\mathrm{poly}(n)$*-time algorithm that outputs all such codewords, then* $\mathcal{C}$ *is said to be* *efficiently* list decodable against t errors*.*

Syndrome decoding is another standard but less common way of characterizing list decodability.

Definition 6 (Syndrome Decoding). *Let* $C : \mathbb{F}_q^k \to \mathbb{F}_q^n$ *be an* $[n,k]_q$ *code with parity check matrix* P*.*

C *is said to be* combinatorially syndrome list decodable against t errors *if for every* $s \in \mathbb{F}_q^{n-k}$*, there are at most* $\mathrm{poly}(n)$ *vectors* $e \in \mathbb{F}_q^n$ *with Hamming weight at most* t *such that* $P \cdot e = s$*.*

C *is said to be* efficiently syndrome list decodable against t errors *if there is a* $\mathrm{poly}(n)$*-time algorithm that on input* $s \in \mathbb{F}_q^{n-k}$ *enumerates all* $e \in \mathbb{F}_q^n$ *with Hamming weight at most* t *for which* $P \cdot e = s$*.*

Fact 1 *An* $[n,k]_q$ *code* C *with parity check matrix* P *is combinatorially list decodable against* t *errors if and only if it is combinatorially* syndrome *list decodable against* t *errors. Moreover* C *is efficiently list decodable against* t *errors if and only if it is efficiently syndrome list decodable against* t *errors.*

Proof. For any $[n,k]_q$ code $C : \mathbb{F}_q^k \to \mathbb{F}_q^n$ with parity check matrix $P \in \mathbb{F}_q^{(n-k)\times n}$, any $t \in \mathbb{Z}^+$, and any $y \in \mathbb{F}_q^n$ with syndrome $s = P \cdot y \in \mathbb{F}_q^{n-k}$, there is a bijective correspondence between:

- $e \in \mathbb{F}_q^n$ such that $P \cdot e = s$ and $\|e\|_0 \leq t$; and
- $m \in \mathbb{F}_q^k$ such that $\|y - C(m)\|_0 \leq t$.

[9] There are multiple possible parity check matrices for any code, but the specific choice will be unimportant for us.

[10] The asymptotic bound of $\mathrm{poly}(n)$ in fact assumes that we have a family of codes for an infinite and dense set of n.

Specifically, any such e can be mapped to $m = C^{-1}(y - e)$. This is well-defined because $P \cdot (y - e) = P \cdot y - P \cdot e = s - s = 0$, so $y - e$ is in the image of C. In the other direction, any such m can be mapped to $e = y - C(m)$, which satisfies $P \cdot e = P \cdot y - P \cdot C(m) = s - 0 = s$. It is easy to check that these maps are inverses of each other.

Known Results

Definition 7 (q-ary Entropy Function). *The q-ary entropy function H_q is defined as*

$$H_q(\rho) \stackrel{\text{def}}{=} \rho \log_q(q-1) - \rho \log_q \rho - (1-\rho)\log_q(1-\rho).$$

We use H without a subscript to refer to the binary entropy function H_2.

We first state the fact that inefficiently (combinatorially) list-decodable codes exist matching the Hamming bound.

Fact 2 (Combinatorially List-Decodable Codes [GHK10]) *For all $q \in \mathbb{Z}^+$, all $0 < \rho < 1 - \frac{1}{q}$, all $0 < R < 1 - H_q(\rho)$, and for all sufficiently large n, there exists an $[n, Rn]_q$ code that is combinatorially list decodable against $\rho \cdot n$ errors. Moreover the list size is inversely proportional to $1 - H_q(\rho) - R$.*[11]

When it comes to efficiently list-decodable codes, we only have construction matching the slightly weaker Blokh-Zyablov bound as stated below.

Fact 3 (Blokh-Zyablov bound [BZ82, GR09]) *For any $q \in \mathbb{Z}^+$, any $\rho \in (0, \frac{1}{2})$, any*

$$0 < R < 1 - \underbrace{\left(H_q(\rho) + \rho \cdot \int_0^{1-H_q(\rho)} \frac{dx}{H_q^{-1}(1-x)} \right)}_{H_q^{\mathsf{BZ}}(\rho)}, \tag{1}$$

and any sufficiently large n, there is an explicit $\left[n, \lceil Rn \rceil\right]_q$ code that is efficiently list decodable against ρn errors.

We define the quantity $H_q^{\mathsf{BZ}}(\rho)$ as in Eq. (1). Note that $\lim_{\rho \to 0} H_q^{\mathsf{BZ}}(\rho) = 0$. In particular, for every $\eta > 0$ there exists some $\rho > 0$ such that $\eta > H_q^{\mathsf{BZ}}(\rho)$.

Lastly, when the distance d is small (say $d \approx n^{\varepsilon}$ for constant $\varepsilon > 0$) then we can get nearly optimal high-rate codes via Reed-Solomon. While Reed-Solomon codes are usually expressed over a large field, if the field is an extension-field of some small field $\mathbb{F}_q$ (e.g., $q = 2$) then we can always re-interpret the codes as just being linear codes over $\mathbb{F}_q$. Therefore Reed-Solomon yields essentially optimal high-rate codes over $\mathbb{F}_2$ when the distance is small.

[11] In fact, a random linear code is known to have the stated list decodability with high probability.

Fact 4 (Efficiently Decodable High-Rate Codes) *For all n, all q, and all d and for $k = n - d\log_q n$ there exists an $\mathbb{F}_q$-linear code $C : \mathbb{F}_q^k \to \mathbb{F}_q^n$ that is efficiently uniquely decodable against $\lfloor \frac{d}{2} \rfloor$ errors.*

Proof. We start with a Reed-Solomon code $C' : (\mathbb{F}')^{k'} \to (\mathbb{F}')^{n'}$, where $n' = n/\log_q n$ and $k' = k/\log_q n$ and $\mathbb{F}' = \mathbb{F}_{q^{\log_q n}}$ is an extension field of $\mathbb{F}_q$ satisfying $|\mathbb{F}'| \geq n \geq n'$. By standard properties of Reed-Solomon codes, C' is $\mathbb{F}'$-linear, has distance $d = n' - k' + 1$, and is efficiently uniquely decodable against $\lfloor d/2 \rfloor$ errors [Pet60].

Using the fact that $\mathbb{F}' \cong \mathbb{F}_q^{\log_q n}$, we can view C' as a code $C : \mathbb{F}_q^k \to \mathbb{F}_q^n$. The code C inherits its distance and efficient unique decodability from C', because if two strings differ in at most d symbols when interpreted as string over $\mathbb{F}_q$, then they also differ in at most d symbols when interpreted as string over $\mathbb{F}'$.

Finally, it is easy to see (e.g., see [RRR21, Proposition 6.6]) that C is $\mathbb{F}_q$-linear.

3 Definition of (R)PPH

We recall the definition of (R)PPH from [BLV19]. We first define a general notion for arbitrary properties P and then discuss the specific Hamming distance property considered in this work. For the general notion, we also potentially consider partial predicates $P(x_1, x_2)$ that can sometimes output $\perp$, in which case we do not care what output the (R)PPH gives.

Definition 8. *Let $n = n(\lambda), m = m(\lambda)$ be some polynomials in the security parameter λ. A (n, m)-Property Preserving Hash (PPH) family $\mathcal{H} = \{h : \{0,1\}^n \to \{0,1\}^m\}$ for a two-input (partial) predicate $P : \{0,1\}^n \times \{0,1\}^n \to \{0,1,\perp\}$ is a family of efficiently computable functions with the following algorithms:*

- $\mathsf{Samp}(1^\lambda) \to h$ *is a PPT algorithm that samples a random $h \in \mathcal{H}$.*
- $\mathsf{Eval}(h, y_1, y_2)$ *is a deterministic polynomial-time algorithm that on input $h \in \mathcal{H}$ and $y_1, y_2 \in \{0,1\}^m$, outputs a single bit.*

Additionally, $h \in \mathcal{H}$ must satisfy the following correctness property:

- *Correctness:* $\forall x_1, x_2 \in \{0,1\}^n$

$$\Pr_{h \leftarrow \mathsf{Samp}(1^\lambda)} [P(x_1, x_2) \neq \perp \wedge \mathsf{Eval}(h, h(x_1), h(x_2)) \neq P(x_1, x_2)] = \mathrm{negl}(\lambda)$$

Definition 9. *A (n, m)-PPH is a* robust PPH *(RPPH) if it satisfies the following additional robustness property.*

- *Robustness: For any PPT adversary $\mathcal{A}$,*

$$\Pr_{\substack{h \leftarrow \mathsf{Samp}(1^\lambda) \\ (x_1, x_2) \leftarrow \mathcal{A}(h)}} [P(x_1, x_2) \neq \perp \wedge \mathsf{Eval}(h, h(x_1), h(x_2)) \neq P(x_1, x_2)] = \mathrm{negl}(\lambda)$$

The main focus of this work is (R)PPH for the (exact) Hamming distance property, which is defined by the following (total) predicate.

Definition 10. *For $n \in \mathbb{N}$, $0 < t < n$, the (two-input)* $\textsc{Hamming}_{n,t}$ *predicate is a predicate defined as*

$$\textsc{Hamming}_{n,t}(x_1, x_2) = \begin{cases} 1 & \text{if } \|x_1 \oplus x_2\|_0 \leq t \\ 0 & \text{if } \|x_1 \oplus x_2\|_0 > t \end{cases}$$

As a tool in one of our constructions, we will also consider a relaxation of (R)PPH to *gap-Hamming distance*, which is defined by the following (partial) predicate.

Definition 11. *For $n \in \mathbb{N}$, $0 < t < n$, $\delta > 0$, the (two-input)* $\textsc{GapHamming}_{n,t,\delta}$ *predicate is a partial predicate defined as*

$$\textsc{GapHamming}_{n,t,\delta}(x_1, x_2) = \begin{cases} 1 & \text{if } \|x_1 \oplus x_2\|_0 \leq t \\ 0 & \text{if } \|x_1 \oplus x_2\|_0 \geq (1+\delta)t \\ \perp & \text{otherwise.} \end{cases}$$

4 Non-robust PPH

In this section, we present the construction of an information-theoretically secure non-robust property preserving hash (PPH) for Hamming distance. The construction relies on syndrome list-decoding and universal hashing.

(n, m)-PPH for Hamming$_{n,t}$.

Let $P \in \mathbb{F}_2^{(n-k)\times n}$ be a parity check matrix of an $[n, k]_2$-linear code which is efficiently list decodable against t errors.

- $\mathsf{Samp}(1^\lambda)$: Sample $A \leftarrow \mathbb{F}_2^{\lambda \times n}$ uniformly at random. Output the function h defined below, whose description contains A.
- $h(x) := (P \cdot x, A \cdot x)$.
- $\mathsf{Eval}(h, y_1, y_2)$: Let $y_1 = (v_1, w_1)$ and $y_2 = (v_2, w_2)$. Use syndrome list-decoding for the syndrome $v_1 - v_2$ to recover a list $\mathcal{L} = \{e_1, \ldots, e_L\}$ of possible error vectors $e_i \in \mathbb{F}_2^n$ such that $Pe_i = v_1 - v_2$ and $\|e_i\|_0 \leq t$. Then
  - output 1, if there exists $e_i \in \mathcal{L}$ such that $A \cdot e_i = w_1 - w_2$,
  - otherwise output 0.

Fig. 1. Construction of (n, m)-PPH for $\textsc{Hamming}_{n,t}$

Theorem 5. *Let λ be a security parameter. For any polynomial n, t, k such that there exists an $[n, k]_2$-linear code which is efficiently list decodable against*

t errors, the construction above is a (n, m)-PPH for $\text{HAMMING}_{n,t}$ *with output length $m = (n - k) + \lambda$. If the code is only combinatorially (inefficiently) list decodable, then the resulting PPH is inefficient.*

Proof. Let $x_1, x_2 \in \mathbb{F}_2^n$ be arbitrary values chosen a priori. Let $h \leftarrow \mathsf{Samp}(1^\lambda)$ and let $y_1 = h(x_1), y_2 = h(x_2)$ with $y_1 = (v_1, w_1)$ and $y_2 = (v_2, w_2)$. Let $\mathcal{L}$ be the list recovered during the computation of $\mathsf{Eval}(h, y_1, y_2)$. We consider two cases:

- If $\|x_1 - x_2\|_0 \leq t$, then $e := x_1 - x_2 \in \mathcal{L}$ by the correctness of syndrome list-decoding for the syndrome $v_1 - v_2 = P \cdot (x_1 - x_2)$. Therefore $A \cdot e = A \cdot (x_1 - x_2) = A \cdot x_1 - A \cdot x_2 = w_1 - w_2$ and hence Eval will output 1.
- If $\|x_1 - x_2\|_0 > t$, then for all $e_i \in \mathcal{L}$, we have $\|e_i\|_0 \leq t$ and therefore $e_i \neq x_1 - x_2$. Note that the list $\mathcal{L}$ is independent of A. Hence for each $e_i \in \mathcal{L}$, we have $\Pr_A[A \cdot e_i = w_1 - w_2] = \Pr_A[A \cdot e_i = A \cdot (x_1 - x_2)] = 2^{-\lambda}$, by Lemma 2. By a union bound, the probability $\Pr_A[\exists e_i \in \mathcal{L} \ : \ A \cdot e_i = w_1 - w_2] \leq |\mathcal{L}| \cdot 2^{-\lambda} = \text{negl}(\lambda)$. Therefore, with all but negligible probability, Eval will output 0.

Plugging in Facts 3 and 4, we obtain the following corollaries for PPH.

Corollary 1. *For all constant $0 < \rho < 1/2$ and $\eta > H_2^{\mathsf{BZ}}(\rho)$ (see Fact 3), for all polynomial n, there exists an efficient (n, m)-PPH for* $\text{HAMMING}_{n,t}$ *with $t = \rho \cdot n$, having output length $m = \eta \cdot n + \lambda$. In particular, for every constant $\eta > 0$, there exists some constant $\rho > 0$ such that the above holds.*

Corollary 2. *For all polynomial n and t, there exists an (n, m)-PPH for* $\text{HAMMING}_{n,t}$ *where $m = 2t \cdot \log_2 n + \lambda$.*

Lastly, plugging in Fact 2, we obtain the following bound for *inefficient* PPH.

Corollary 3. *For all constant $0 < \rho < 1/2$ and $\eta > H_2(\rho)$ there exists an* inefficient *(n, m)-PPH for* $\text{HAMMING}_{n,t}$ *with $t = \rho \cdot n$, having output length $m = \eta \cdot n + \lambda$.*

5 Lower Bounds on PPH Output

In this section we provide a lower bound on the output size of any (not necessarily robust) PPH for the Hamming distance predicate. Previous lower bounds on PPH [BLV19, FLS22] mainly come from communication complexity lower bounds, and are usually presented as asymptotic bounds. In particular, in [FLS22] the authors presented an output size bound of $\Omega(t \log(\min n/t, 1/\delta))$ for RPPHs for $\text{HAMMING}_{n,t}$ with error probability δ. We obtain an exact lower bound, without asymptotic. In particular, this lets us argue that we cannot beat the Hamming bound.

As with all previous lower bounds on RPPH ([BLV19, FLS22]), our lower bound works for non-robust PPH as well. It remains unclear how robustness is factored into RPPH lower bounds.

Theorem 6. *For any (n, m)-PPH for $\text{HAMMING}_{n,t}$ with correctness error $\delta < \frac{1}{2n}$, the output size m must satisfy $m \geq \log \binom{n}{t}$. In particular, this implies $m \geq t \log \frac{n}{t}$ and $m \geq (H\left(\frac{t}{n}\right) - o(1)) \cdot n$, where H is the binary entropy function.*

Proof. Let $(\mathsf{Samp}, \mathsf{Eval})$ be an (n, m)-PPH for $\text{HAMMING}_{n,t}$. We first show that there is some function Rec such that, for all x, y with $||x - y||_0 = t$, we have

$$\Pr_{h \leftarrow \mathsf{Samp}()}[\mathsf{Rec}(h, h(x), y) = x] > \frac{1}{2}.$$

In particular, we define $\mathsf{Rec}(h, h(x), y)$ as follows:

- For each $i \in [n]$, define the string $y^{(i)}$ to be the same as y except that we flip the i'th bit. Then compute $b_i = \mathsf{Eval}(h, h(x), h(y^{(i)}))$. Let $\tilde{x}_i = b_i \oplus y_i$, where y_i denotes the i'th bit of y.
- Output $\tilde{x} = (\tilde{x}_1, \ldots, \tilde{x}_n)$.

Observe that for each $i \in [n]$, if $x_i = y_i$, we have $\left\|x - y^{(i)}\right\|_0 = t + 1$ and if $x_i \neq x'_i$, we have $\left\|x \oplus y^{(i)}\right\|_0 = t - 1$. Therefore, as long as for each $i \in [n]$, we get $b_i = 0$ in the former case and $b_i = 1$ in the latter case, we get $\tilde{x} = x$. By the correctness of the PPH, this occurs with overwhelming probability, since the probability of Eval giving the wrong answer in any position i is $< \frac{1}{2n}$, and by the union bound, the probability of there being any error overall is then $< \frac{1}{2}$.

Now let us define a (randomized) function $P(h, h(x))$ whose goal is to predict x given $h, h(x)$:

- Sample a uniformly random $y \leftarrow \{0,1\}^n$ and output $\mathsf{Rec}(h, h(x), y)$,

We have for all $x \in \{0,1\}^n$:

$$\begin{aligned}
&\Pr_{h \leftarrow \mathsf{Samp}()}[P(h, h(x)) = x] \\
&\geq \Pr_{y \leftarrow \{0,1\}^n}[\ ||x - y||_0 = t\ ] \Pr_{h,y}[\ \mathsf{Rec}(h, h(x), y) = x \ \mid\ ||x - y||_0 = t\ ] \\
&> \frac{\binom{n}{t}}{2^n} \cdot \frac{1}{2}.
\end{aligned}$$

Now consider X to be a random variable uniformly distributed over $\{0,1\}^n$ and h to be a random variable distributed according to $\mathsf{Samp}()$. Then

$$\mathbf{H}_\infty(X \mid h, h(X)) \leq -\log(\mathbf{Pred}(X \mid h, h(X))) \leq -\log(\Pr_{h,x}[P(h, h(x)) = x]) < n - \log \binom{n}{t} + 1.$$

On the other hand

$$\mathbf{H}_\infty(X \mid h, h(X)) \geq \mathbf{H}_\infty(X \mid h) - m \geq \mathbf{H}_\infty(X) - m \geq n - m.$$

The first inequality follows from Lemma 1, the second inequality follows since h, X are independent, and the third since X is uniformly random over $\{0,1\}^n$.

Combining the above two inequalities, we get $m > \log \binom{n}{t} - 1$, but since m is an integer, this implies $m \geq \log \binom{n}{t}$.

Optimality of Our Construction. Our lower bound in Theorem 6 shows that our PPH construction from the previous section achieves essentially tight parameters. We look at two concrete settings.

When $t = \rho \cdot n$ for some constant $\rho > 0$ then the lower bound shows $m \geq (H(\rho) - o(1)) \cdot n$. This essentially matches our upper bound from Corollary 3 which shows that inefficiently we can achieve $m \approx H(\rho) \cdot n + \lambda$. Therefore, our inefficient construction is tight, including constant factors! Efficiently, Corollary 1 allows us to achieve a slightly worse compression $m \approx H_2^{\mathsf{BZ}}(\rho) \cdot n + \lambda$, where $H_2^{\mathsf{BZ}}(\rho)$ is a slightly larger constant than $H(\rho)$. The small gap between our efficient upper bounds and the lower bound is due to the fact that current constructions of efficiently list-decodable linear codes are slightly sub-optimal compared to combinatorially (inefficiently) list-decodable counterparts – future advances in coding theory will hopefully allows us to close this gap.

For smaller distances $t = n^{\varepsilon}$ for some constant $\varepsilon > 0$, then the lower bound shows that $m \geq t \log(n/t) = \Omega(t \log n)$. This essentially matches our upper bound of $m = O(t \log n)$ from Corollary 2, up to constant factors.

6 RPPH from Homomorphic Collision Resistance

In this section, we present the construction of a robust property preserving hash (RPPH) from homomorphic collision resistant hash functions. The construction is analogous to that of non-robust PPH, with the main difference being that we replace the homomorphic universal hash function $h_{univ}(x) = A \cdot x$ with a cryptographic homomorphic collision-resistant hash function. We begin by defining this notion and showing how to construct it.

6.1 Homomorphic CRHFs

We rely on the following definition of a homomorphic collision-resistant hash function.

Definition 12 (Homomorphic CRHF). *Let $n = n(\lambda), \ell = \ell(\lambda)$ be some polynomials with $\ell < n$. A family of* Homomorphic Collision Resistant Hash Functions $(\mathsf{Samp}, \mathcal{H})$ *consists of a sampling algorithm $h \leftarrow \mathsf{Samp}(1^\lambda)$ that generates a hash function $h \in \mathcal{H}$ with $h : \mathbb{Z}_q^n \to \{0,1\}^\ell$ for some integer q specified by h. We require require the following properties:*

- Efficiency: *For any $h \leftarrow \mathsf{Samp}(1^\lambda)$ the function $h(x)$ can be computed in* $\mathrm{poly}(\lambda)$ *time.*
- Collision-Resistance: *For any ppt adversary $\mathcal{A}$:*

$$\Pr\left[\begin{array}{l} h(x_1) = h(x_2) \\ \wedge \quad (x_1 \neq x_2) \\ \wedge \quad x_1, x_2 \in \{-1, 0, 1\}^n \end{array} \;:\; \begin{array}{r} h \leftarrow \mathsf{Samp}(1^\lambda), \\ (x_1, x_2) \leftarrow \mathcal{A}(h) \end{array}\right] \leq \mathrm{negl}(\lambda).$$

- Homomorphism: *The description of h determines some operation $\div$ computable in* $\mathrm{poly}(\lambda)$ *time such that for all $x_1, x_2 \in \mathbb{Z}_q^n$ we have*

$$h(x_1) \div h(x_2) = h(x_1 - x_2).$$

Note that, while for homomorphism we consider the domain of the hash function to be $\mathbb{Z}_q^n$ and the subtraction $x_1 - x_2$ is computed in $\mathbb{Z}_q^n$, for collisions we only need to consider inputs in a restricted sub-domain $\{-1, 0, 1\}^n \subseteq \mathbb{Z}_q^n$. This will be important for our construction from the SIS assumption.

Construction from Discrete Log. We observer that the Pedersen hash function [Ped92] (a deterministic version of Pedersen commitment) is a good homomorphic collision-resistant hash function under the discrete logarithm assumption.

Let $\mathcal{G} = (\mathbb{G}, g, q) \leftarrow \mathsf{GroupGen}(1^\lambda)$ be a group generation algorithm that generates the description of a cyclic group $\mathbb{G} = \langle g \rangle$ of prime order $|\mathbb{G}| = q$, with a generator g, such that the group operation (written as multiplication) can be computed in $\mathrm{poly}(\lambda)$ time and group elements can be efficiently represented using $\ell = \ell(\lambda)$ bits.

The discrete logarithm (DLOG) assumption relative to the above $\mathsf{GroupGen}$ algorithm says the following.

Definition 13 (Discrete Log Assumption.). *For any ppt adversary $\mathcal{A}$ we have:*

$$\Pr[\mathcal{A}(\mathcal{G}, g^x) = x \ : \ \mathcal{G} \leftarrow \mathsf{GroupGen}(1^\lambda), x \leftarrow \mathbb{Z}_q] \leq \mathrm{negl}(\lambda).$$

For any polynomial input length $n = n(\lambda)$, the Pederson hash functions $(\mathsf{Samp}, \mathcal{H})$ is defined as follows:

- $h \leftarrow \mathsf{Samp}(1^\lambda)$: Sample $\mathcal{G} = (\mathbb{G}, g, q) \leftarrow \mathsf{GroupGen}(1^\lambda)$, and let $g_1, \ldots, g_n \leftarrow \mathbb{G}$ be random group elements. The description of the hash function h consists of $(\mathcal{G}, g_1, \ldots, g_n)$.
- $y = h(x)$: Given an input $x = (x_1, \ldots, x_n) \in \mathbb{Z}_q^n$ define $h(x) = \prod_{i \in [n]} g_i^{x_i}$.
- The $\div$ operation is defined as $h(x) \div h(x') = h(x)/h(x') = h(x - x')$.

Theorem 7 ([Ped92]). *The above hash function family is a homomorphic collision-resistant hash function under the discrete logarithm assumption.*

Construction from SIS. We observe that Ajtai's hash function [Ajt96] based on the short-integer solution (SIS) problem is is a good homomorphic collision-resistant hash function.

Definition 14. *The short integer solution $SIS_{m,q,B}$ assumption with some parameters $m = m(\lambda), q = q(\lambda)$ and $B = B(\lambda)$ says that for all polynomial $n = n(\lambda)$ and all ppt $\mathcal{A}$ we have:*

$$\Pr[A \cdot x = 0 \wedge x \neq 0 \wedge x \in [-B, B]^n \ : \ A \leftarrow \mathbb{Z}_q^{m \times n}, x \leftarrow \mathcal{A}(A)] \leq \mathrm{negl}(\lambda).$$

For any polynomial input length $n = n(\lambda)$, Ajtai's hash functions $(\mathsf{Samp}, \mathcal{H})$ is defined as follows:

- $h \leftarrow \mathsf{Samp}(1^\lambda)$: Sample $A \leftarrow \mathbb{Z}_q^{m \times n}$ and let the description of the hash function $h : \mathbb{Z}_q^n \to \mathbb{Z}_q^m$ consist of the matrix A.
- $y = h(x)$: Given an input $x \in \{0, 1\}^n$ define $h(x) = A \cdot x$.

– The $\div$ operation is defined as $h(x_1) \div h(x_2) = h(x_1) - h(x_2)$.

Theorem 8 ([Ajt96]). *The above hash function family is a homomorphic collision-resistant hash function under the $SIS_{m,q,B}$ assumption with $B = 2$.*

Proof. Assume otherwise, that there is some ppt $\mathcal{A}$ such that

$$\Pr[A{\cdot}x_1 = A{\cdot}x_2 \wedge x_1 \neq x_2 \wedge x_1, x_2 \in \{-1, 0, 1\}^n : A \leftarrow \mathbb{Z}_q^{m \times n}, (x_1, x_2) \leftarrow \mathcal{A}(A)] = \mu(\lambda)$$

for some non-negligible μ. Whenever $A{\cdot}x_1 = A{\cdot}x_2 \wedge x_1 \neq x_2 \wedge x_1, x_2 \in \{-1, 0, 1\}^n$ occurs, we can define $x^* = (x_1 - x_2) \in [-2, 2]^n$ such that $x^* \neq 0$ and $Ax^* = 0$. Therefore if we define a ppt $\mathcal{A}'$ that runs $(x_1, x_2) \leftarrow \mathcal{A}(A)$ and outputs $x^* = (x_1 - x_2)$ then

$$\Pr[A \cdot x^* = 0 \wedge x^* \neq 0 \wedge x^* \in [-2, 2]^n \ : \ A \leftarrow \mathbb{Z}_q^{m \times n}, x^* \leftarrow \mathcal{A}'(A)] = \mu(\lambda).$$

and therefore $\mathcal{A}'$ breaks the $SIS_{m,q,B}$

6.2 RPPH from Homomorphic CRHFs

We now give our construction of RPPH from homomorphic CRHFs. The construction is essentially identical to the non-robust PPH construction in Figure 1. The main difference is that we now use a homomorphic CRHF in place of a homomorphic universal hash function. Another difference arises from the fact that the homomorphic CRHF is over $\mathbb{Z}_q^n$ for some arbitrary q rather than just over $\mathbb{Z}_2$. Although we will still apply the CRHF on inputs $x_1, x_2 \in \{0,1\}^n$, when we subtract over $\mathbb{Z}_q^n$ we get $x_1 - x_2 \in \{-1, 0, 1\}^n$. If $q \neq 2$ then $-1 \neq 1$. This means that we need to use a linear error-correcting code over some field $\mathbb{F}$ of characteristic $p > 2$ so that when we apply syndrome decoding over $\mathbb{F}$ we correctly recover the same value $x_1 - x_2 \in \{-1, 0, 1\}^n$.

Theorem 9. *Assume the existence of a homomorphic CRHF with output length $\ell(\lambda)$. For any polynomial t, k and odd prime power Q such that there exists an $[n, k]_Q$ code that is efficiently list decodable against t errors, the construction above is an (n, m)-RPPH for $\textsc{Hamming}_{n,t}$ with output length $(n - k) \cdot \log_2 Q + \ell(\lambda)$.*

Proof. Let $h \leftarrow \mathsf{Samp}(1^\lambda)$, let $x_1, x_2 \in \{0,1\}^n$ be arbitrary values chosen by an adversary adaptively after seeing h. Let $y_1 = h(x_1), y_2 = h(x_2)$ with $y_1 = (v_1, w_1)$ and $y_2 = (v_2, w_2)$. Let $\mathcal{L}$ be the list recovered during the computation of $\mathsf{Eval}(h, y_1, y_2)$. We consider two cases:

– If $\|x_1 - x_2\|_0 \leq t$, then $e := x_1 - x_2 \in \mathcal{L}$ by the correctness of syndrome list-decoding for the syndrome $v_1 - v_2 = P \cdot (x_1 - x_2)$. Therefore $g(e) = g(x_1 - x_2) = g(x_1) \div g(x_2) = w_1 \div w_2$ and hence Eval will output 1. Note that, since $x_1, x_2 \in \{0,1\}^n$, the difference $x_1 - x_2 \in \{-1, 0, 1\}^n$ is the same when computed over the field $\mathbb{F}$ of characteristic $p \geq 3$ as when just computed over the integers.

(n, m)-RPPH for Hamming$_{n,t}$.

Let $n = n(\lambda)$ and $\ell = \ell(\lambda)$. Let $P \in \mathbb{F}^{(n-k)\times n}$ be a parity check matrix of an $[n, k]_Q$ code that is efficiently list decodable against t errors, where Q is an odd prime power. Let $(\mathsf{Samp}_{CR}, \mathcal{H}_{CR})$ be a family of collision Resistant Homomorphic Hash Functions with input length n and output length ℓ. The RPPH family $(\mathsf{Samp}, \mathsf{Eval})$ is defined as follows:

- $\mathsf{Samp}(1^\lambda)$: Sample $g \leftarrow \mathsf{Samp}_{CR}(1^\lambda)$ to generate a homomorphic collision-resistant hash function $g : \mathbb{Z}_q^n \to \{0,1\}^\ell$ for some q. Output the function h defined below, whose description contains g.
- $h(x) := (P \cdot x, g(x))$.
- $\mathsf{Eval}(h, y_1, y_2)$: Let $y_1 = (v_1, w_1)$ and $y_2 = (v_2, w_2)$. Use syndrome list-decoding for the syndrome $v_1 - v_2 \in \mathbb{F}^{n-k}$ to recover a list $\mathcal{L} = \{e_1, \ldots, e_L\}$ of possible error vectors $e_i \in \mathbb{F}^n$ such that $Pe_i = v_1 - v_2$ and $\|e_i\|_0 \le t$. Then
  - output 1, if there exists $e_i \in \mathcal{L}$ such that $g(e_i) = w_1 \div w_2$ and $e_i \in \{-1, 0, 1\}^n$,
  - otherwise output 0.

Fig. 2. Construction of (n, m)-RPPH for Hamming$_{n,t}$.

- If $\|x_1 - x_2\|_0 > t$, then for all $e_i \in \mathcal{L}$, we have $\|e_i\|_0 \le t$ and therefore $e_i \neq x_1 - x_2 \pmod q$. If there exists some i such that $e_i \in \{-1, 0, 1\}^n$ and $g(e_i) = g(x_1 - x_2)$ it means that we found a valid collision $e_i \neq (x_1 - x_2)$ in the hash function g. But, by collision resistance, the probability of this is negligible. Therefore, with overwhelming probability, no such index i exists and Eval will output 0.

Plugging in Facts 3 and 4, and using $Q = 3$, we obtain the following corollaries for PPH.

Corollary 4. *Assume the existence of a homomorphic CRHF with output length $\ell = \ell(\lambda)$. For all constants $0 < \rho < 1/2$ and $\eta > H_3^{\mathsf{BZ}}(\rho) \cdot (\log_2 3)$ (see Fact 3), for all polynomial n, there exists an efficient (n, m)-PPH for Hamming$_{n,t}$ with $t = \rho \cdot n$, having output length $m = \eta \cdot n + \ell$. In particular, for every constant $\eta > 0$, there exists some constant $\rho > 0$ such that the above holds.*

Corollary 5. *Assume the existence of a homomorphic CRHF with output length $\ell = \ell(\lambda)$. For all polynomial n and t, there exists an (n, m)-RPPH for Hamming$_{n,t}$ where $m = O(t \cdot \log n + \ell)$.*

In the case where $t = \rho \cdot n$ for a constant $\rho > 0$, the first corollary gives constant compression factor $\eta \approx H_3^{\mathsf{BZ}}(\rho) \cdot (\log_2 3)$. We know from our lower bound (Theorem 6) that we cannot do better than $\eta > H(\rho)$. The small gap between the constant in our upper bounds and lower bounds comes from: (1) the fact that current constructions of efficiently list-decodable linear codes are

slightly sub-optimal compared to combinatorially (inefficiently) list-decodable counterparts, which reuslts in our upper bound having H^{BZ} instead of H, (2) the fact that our constructions of homomorphic hash functions work over $\mathbb{Z}_q$ for $q > 2$ rather than over $\mathbb{Z}_2$, which necessitates the $\log_2 3$ factor. It is plausible that future advances in list-decodable codes and homomorphic hashing could remove either/both of these gaps.

In the case of smaller distances t in the range $\ell < t < n^{\varepsilon}$ for some constant $\varepsilon > 0$, the lower bound (Theorem 6) shows that $m \geq t \log(n/t) = \Omega(t \log n)$, which essentially matches our upper bound of $m = O(t \log n)$, up to constant factors.

7 RPPH from Standard Collision Resistance

In this section, we present our construction of RPPH for exact Hamming distance from standard collision-resistant hash functions. We do so by starting with the construction of RPPH for gap Hamming distance due to Boyle, LaVigne, and Vaikuntanathan in [BLV19], and then showing how to generically upgrade an RPPH for gap Hamming to an RPPH for exact Hamming using syndrome decoding.

7.1 RPPH for Gap-Hamming

We start with a RPPH construction for $\textsc{GapHamming}_{n,t,\delta}$ described by Boyle, LaVigne, and Vaikuntanathan in [BLV19]. We give a generalized (and somewhat simplified) analysis of the construction that explicitly shows how the output length m scales as a function of the input length n, the distance t and the security parameter λ for general setting of parameters.

Theorem 10 (Generalizing [BLV19] Theorem 16). *Let λ be a security parameter and let $\delta > 0$ be a constant. Assuming CRHFs with output size $\ell = \mathrm{poly}(\lambda)$, for any $n = \mathrm{poly}(\lambda)$, any $0 < t < n$, there exists a (n, m)-RPPH for $\textsc{GapHamming}_{n,t,\delta}$ with output length $m = O((t \log \frac{n}{t} + \lambda)\ell)$.*

The main idea of the construction is to use a bipartite "expander graph" to map the n locations of the input into k subsets for some $k \ll n$. Then, we apply a standard CRHF on the bits of x in each of the k sets of locations and set the k CRHF outputs as the output of the RPPH. The expander ensures that there is some threshold μ such that:

- If x_1, x_2 differ in $\leq t$ locations then they will differ $< \mu \cdot k$ of the k subsets and therefore at most that many of the CRHF output will differ.
- If x_1, x_2 differ in $> (1 + \delta)t$ locations then they will differ $> \mu \cdot k$ of the k subsets and therefore at least that many of the CRHF output will differ.

This allows us to distinguish the two cases.

In [BLV19], they rely on standard expander graphs to achieve the above properties. We observe that the expansion is somewhat stronger than what we

need: it guarantees that for *every* small set $S \subseteq [n]$, must have a large number of neighbors, whereas we only need this to hold when $|S| \geq (1+\delta)t$. As a result, we obtain a more straightforward analysis for the same construction and a more general range of parameters.

Definition 15. *Let $G = (L \cup R, E)$ be a bipartite graph with $E \subseteq L \times R$. For a set $S \subseteq L$ let $N(S) \subseteq R$ denote the neighbors of S. We say that G is (n, k, t, δ)-nice if it has $|L| = n, |R| = k$, and there exists some threshold $\mu > 0$ such that:*

1. *For every "small" $S \subseteq L$ such that $|S| \leq t$, we have $|N(S)| < \mu k$.*
2. *For every "large" $S \subseteq L$ such that $|S| \geq (1+\delta)t$ we have have $|N(S)| > \mu k$.*

We show that such nice graphs exist and can be sampled efficiently via a probabilistic argument. Indeed, a random graph is nice with overwhelming probability. We can rely on randomized constructions since we can include the description of the graph G as part of the description of the RPPH.

Lemma 3. *For any n, $0 < t < n$, and any constant $\delta > 0$, a (n, k, t, δ)-nice bipartite graph is efficiently constructible (with all but $e^{-\Omega(\lambda)}$ probability) with $k = O(t \log(n/t) + \lambda)$.*

Proof. We show that a random graph satisfies the requirement with all but negligible probability. Define the following constants that depend on δ:

$$\mu_0 = \frac{\delta}{2(1+\delta)^2}, \quad \mu_1 = (1+\delta/2)\mu_0, \quad \rho = \frac{\delta}{4+\delta}, \quad \mu = (1+\rho)\mu_0 = (1-\rho)\mu_1.$$

Sample a bipartite graph $G = (L \cup R, E)$ with $|L| = n$, $|R| = k$, where for any $(v, w) \in L \times R$, the edge (v, w) is included in E independently with probability $p = \frac{\mu_0}{t}$. We show that this graph is (n, k, t, δ)-nice with overwhelming probability. We do so by showing that each of the two properties holds separately.

First, we show property 1 holds. Fix any set $S \subseteq L$ of size $|S| = t$. For any $w \in R$, we can rely on the union bound to show:

$$\Pr[w \in N(S)] = \Pr\left[\bigcup_{v \in S} \ (v, w) \in E \ \right] \leq \sum_{v \in S} \Pr[(v, w) \in E] \leq t \cdot p \leq \mu_0.$$

Define the indicator random variables X_w which are 1 iff $w \in N(S)$. Then these random variables are independent and $\mathbb{E}[\sum_{w \in R} X_w] = \mu_0 \cdot k$. By the Chernoff bound, we therefore have:

$$\Pr\left[\ |N(S)| \geq \mu \cdot k\ \right] = \Pr\left[\ \sum_{w \in R} X_w \geq (1+\rho)\mu_0 k\ \right] \leq \exp\left(-\rho^2 \mu_0 k/3\right).$$

Finally, by the union bound over all such sets S, we can bound the probability that property 1 does *not* hold by:

$$\Pr\left[\exists S \subseteq L, |S| = t \ : \ |N(S)| \geq \mu \cdot k\right] \leq \binom{n}{t} \cdot \exp\left(-\rho^2 \mu_0 k/3\right).$$

By choosing a sufficiently large $k = O(t \log \frac{n}{t} + \lambda)$, we can bound the above by $2^{-\Omega(\lambda)}$.

Second, we show property 2 holds. Fix any set $S \subseteq L$ of size $|S| \geq (1+\delta)t$. For any $w \in R$, we can rely on the inclusion-exclusion principle to show:

$$\begin{aligned}
\Pr[w \in N(S)] &= \Pr\left[\bigcup_{v \in S} \ (v,w) \in E \ \right] \\
&\geq \sum_{v \in S} \Pr[(v,w) \in E] - \sum_{v_1 \neq v_2 \in S} \Pr[(v_1,w) \in E \wedge (v_2,w) \in E] \\
&\geq (1+\delta)tp - [(1+\delta)t]^2 p^2 \\
&= (1+\delta)tp - (\delta/2)tp \\
&= (1+\delta/2)tp = \mu_1.
\end{aligned}$$

Define the indicator random variables X_w which are 1 iff $w \in N(S)$. Then these random variables are independent and $\mathbb{E}[\sum_{w \in R} X_w] = \mu_1 \cdot k$. By the Chernoff bound, we therefore have:

$$\Pr\left[\ |N(S)| \leq \mu \cdot k\ \right] = \Pr\left[\ \sum_{w \in R} X_w \leq (1-\rho)\mu_1 k\ \right] \leq \exp\left(-\rho^2 \mu_0 k/2\right).$$

Finally, by the union bound over all such sets S, we can bound the probability that property 2 does *not* hold by:

$$\Pr\left[\exists S \subseteq L, |S| = (1+\delta)t \ : \ |N(S)| \geq \mu \cdot k\right] \leq \binom{n}{(1+\delta)t} \exp\left(-\rho^2 \mu_0 k/2\right).$$

By choosing a sufficiently large $k = O(t \log \frac{n}{t} + \lambda)$, we can bound the above by $e^{-\Omega(\lambda)}$.

Therefore, for each property, the probability that it fails to hold is negiligible, and by the union bound, the probability that either property fails to hold is then also negligible. This shows that the samples graph G is (n, k, t, δ)-nice with all but $e^{-\Omega(\lambda)}$ probability.

The proof of Theorem 10 is similar to the proof of Theorem 16 in [BLV19], modulo parameter settings and the graph G defined above. We present the proof for completeness.

Proof (Proof of theorem 10). We show that the construction in Figure 3 yields an RPPH construction. We split the proof of robustness into two cases:

- Suppose $x_1, x_2 \in \{0,1\}^n$ satisfy $\|x_1 \oplus x_2\|_0 \leq t$. Let $S \subseteq L$ be the set of indices where x_1, x_2 differ, and $T = N(S)$. We have $|S| \leq t$. Since G is nice with overwhelming probability, by the first property we have $|T| < \mu k$ with overwhelming probability.

(n, m)-**RPPH for GapHamming**$_{n,t,\delta}$

Given parameters $n, 0 < t < n, \delta > 0$ and a CRHF family $(\mathsf{Samp}_{CR}, \mathcal{G})$:

- $\mathsf{Samp}(1^\lambda)$: Set $k = O(t \log \frac{n}{t} + \lambda)$ appropriately and sample a (n, k, t, δ)-nice bipartite graph $G = (L \cup R, E)$ per Lemma 3. Sample a CRHF $g \leftarrow \mathsf{Samp}_{CR}(1^\lambda)$ with output size $\ell = \ell(\lambda)$. Output $h = (G, g)$.
- $h(x)$: For every $j \in [k]$, compute the (ordered) set of neighbors of the j-th right vertex in G, denoted $N(j)$. Let $\hat{x}^{(j)} := x|_{N(j)}$ be x restricted to the set $N(j)$. Output $h(x) := g(\hat{x}^{(1)}), ..., g(\hat{x}^{(k)})$.
- $\mathsf{Eval}(h = (G, g), y_1, y_2)$. Parse $y_1 = (\hat{y}_1^{(1)}, \ldots, \hat{y}_1^{(k)})$ and $y_2 = (\hat{y}_2^{(1)}, \ldots, \hat{y}_2^{(k)})$. Compute

$$\Delta = \sum_{j=1}^{k} \mathbb{1}(\hat{y}_1^{(j)} \neq \hat{y}_2^{(j)})$$

  Let μ be the threshold parameter for the (n, k, t, δ)-nice bipartite graph G per definition 15. If $\Delta \leq \mu k$, output 1. Otherwise, output 0.

Fig. 3. Construction of (n, m)-RPPH family for $\textsc{GapHamming}_{n,t,\delta}$ from CRHFs

Now for every $j \in R$ such that $j \notin T$, the subsampled values satisfy $\hat{x}_1^{(j)} = \hat{x}_2^{(j)}$ and in turn $\hat{y}_1^{(j)} = \hat{y}_2^{(j)}$. Therefore

$$\Delta = \sum_{j=1}^{k} \mathbb{1}(\hat{y}_1^{(i)} \neq \hat{y}_2^{(i)}) \leq |T| < \mu k$$

so Eval will output 1 with overwhelming probability.

– Now suppose $x_1, x_2 \in \{0,1\}^n$ satisfy $\|x_1 \oplus x_2\|_0 \geq (1+\delta)t$. Define S, T as above, then $|S| \geq (1+\delta)t$. Since G is nice, by the second property we have $|T| > \mu k$, with overwhelming probability.
Now for every $j \in T$, $\hat{x}_1^{(j)} \neq \hat{x}_2^{(j)}$. We show that $\hat{y}_1^{(j)} \neq \hat{y}_2^{(j)}$ with all but negligible probability for (x_1, x_2) chosen by a PPT adversary. Suppose not, then the $\hat{x}_1^{(j)}, \hat{x}_2^{(j)}$ are a collision on the CHRF, which contradicts collision resistance.
Therefore with all but negligible probability for each $j \in T$ we have $\hat{y}_1^{(j)} \neq \hat{y}_2^{(j)}$. Using a union bound this holds for all $j \in T$ still with all but negligible probability, in which case $\Delta > \mu \cdot k$ and Eval will output 0.

7.2 From Gap-Hamming to Hamming

Now we are ready to use syndrome decoding to generically amplify any RPPH for gap Hamming distance to a RPPH for exact Hamming distance. In this section, we rely on unique decoding rather than list decoding.

(n, m)-**RPPH for Hamming**$_t$

Given n, $0 < t < n$: Let $\delta > 0$ be any constant. Fix a (n, m_{gap})-RPPH $(\mathsf{Samp}_{\mathsf{gap}}, \mathsf{Eval}_{\mathsf{gap}})$ for $\textsc{GapHamming}_{n,t,\delta}$ with output size m_{gap} and a parity-check matrix $P \in \mathbb{F}_2^{(n-k)\times n}$ of a (n, k)-linear code that allows efficient (unique) decoding up to $(1 + \delta)t$ errors.

- $\mathsf{Samp}(1^\lambda)$: Sample $g \leftarrow \mathsf{Samp}_{\mathsf{gap}}(1^\lambda)$. Output the hash function h defied below, whose description consists of the description of g.
- $h(x) := (Px, g(x))$.
- $\mathsf{Eval}(h, y_1, y_2)$: Parse $y_1 = (Px_1, g(x_1))$ and $y_2 = (Px_2, g(x_2))$. Let $s := Px_1 \oplus Px_2 = P(x_1 \oplus x_2)$. Apply syndrome decoding to obtain e such that $P \cdot e = s$ and $\|e\|_0 \leq (1+\delta)t$, or output 0 if no such e exists. Set $b = \mathsf{Eval}_{\mathsf{gap}}(g, g(x_1), g(x_2))$. Then
  - output 1 if $\|e\|_0 \leq t$ and $b = 1$.
  - output 0 otherwise.

Fig. 4. Construction of RPPH for $\textsc{Hamming}_{n,t}$ from RPPH for $\textsc{GapHamming}_{n,t,\delta}$

Theorem 11. *Let λ be a security parameter and $\delta > 0$ be any constant. Assume there exist a (n, m_{gap})-RPPH for $\textsc{GapHamming}_{n,t,\delta}$ with output size m_{gap} and an efficiently decodable (n, k)-linear code that corrects up to $(1+\delta)t$ errors. Then the construction in figure 4 is a (n, m)-RPPH for $\textsc{Hamming}_{n,t}$ with output length $m = n - k + m_{gap}$.*

Proof. We analyze robustness. Suppose a PPT adversary $\mathcal{A}$ outputs $x_1, x_2 \in \{0, 1\}^n$:

- If x_1, x_2 satisfy $\|x_1 - x_2\|_0 \leq t$, then syndrome decoding of $s = P \cdot (x_1 - x_2)$ always recovers $e = x_1 - x_2$, since $\|e\|_0 \leq t$. Now since g is a RPPH for $\textsc{GapHamming}_{n,t,\delta}$, by its robustness we have $\mathsf{Eval}_{\mathsf{gap}}(g, g(x_1), g(x_2)) = 1$ with probability $1 - \mathrm{negl}(\lambda)$. Therefore $\mathsf{Eval}(h, h(x_1), h(x_2))$ will output 1 with probability $1 - \mathrm{negl}(\lambda)$.
- If x_1, x_2 satisfy $\|x_1 - x_2\|_0 \geq t+1$, we further distinguish between two cases: (1) If $t + 1 \leq \|x_1 - x_2\|_0 \leq (1 + \delta)t$, syndrome decoding always recovers $e = (x_1 - x_2)$ and $\|e\|_0 > t$ so Eval will output 0 with probability 1. (2) If $\|x_1 \oplus x_2\|_0 > (1 + \delta)t$, by the robustness of g, $\mathsf{Eval}_{\mathsf{gap}}(g, g(x_1), g(x_2)) = 0$ with probability $1 - \mathrm{negl}(\lambda)$, so Eval will output 0 with probability at least $1 - \mathrm{negl}(\lambda)$.

Plugging in Theorem 10 along with Fact 4 to the above, and setting (e.g.,) $\delta = 1$, gives the following corollary.

Corollary 6. *Assume there exists a CRHF with output length $\ell = \ell(\lambda)$. For any polynomial n and t, there exists an (n, m)-RPPH for $\textsc{Hamming}_{n,t}$ with output length $m = O(\ell \cdot t \cdot \log(n/t) + \ell \cdot \lambda)$.*

The above essentially matches the parameters of prior works [FS21, FLS22] that relied on specific algebraic asssumptions: the q-Strong Bilinear Discrete Logarithm (q-SBDL) Assumption in the former case, and the Short-Integer Solution

(SIS) in the latter case. It also essentially matches (a generalized form of) the parameters achieved by the prior work of [BLV19] for gap Hamming distance, but does so for exact Hamming distance.

7.3 The Necessity of Collision-Resistance

We also show that collision-resistant hash functions are necessary for RPPH for Hamming distance, and therefore our construction above is based on a minimal assumption. This result follow implicitly as a special case of a result of [BLV19] (Corollary 30), but we include a simple stand-alone proof for completeness.

Theorem 12. *Let $n = n(\lambda)$ and $t = t(\lambda)$ be polynomials with $t < n/2$. Any (n, m)-RPPH for $\text{HAMMING}_{n,t}$ is also necessarily a collision-resistant hash function.*

Proof. First observe that for any $x_1, x_2 \in \{0,1\}^n$ with $x_1 \neq x_2$ we can efficiently find y such that $||x_1 - y||_0 \leq t$ and $||x_2 - y||_0 > t$. In particular, if $||x_1 - x_2||_0 > t$ then $y = x_1$ satisfies this, and otherwise define y by flipping some arbitrary t positions of x_1 in which x_1, x_2 agree.

Now assume we have an RPPH construction which is not collision-resistant, meaning that there is some PPT $\mathcal{A}$ such that, given $h \leftarrow \mathsf{Samp}(1^\lambda)$, the output $(x_1, x_2) \leftarrow \mathcal{A}(h)$ is a valid collision with non-negligible probability, meaning: $x_1 \neq x_2$ and $h(x_1) = h(x_1)$. Whenever $\mathcal{A}(h)$ finds a valid collision (x_1, x_2), we can use it to find inputs on which the RPPH give the wrong answer. Namely, we can find y as above with $||x_1 - y||_0 \leq t$ and $||x_2 - y||_0 > t$. Then $\text{HAMMING}_{n,t}(x_1, y) \neq \text{HAMMING}_{n,t}(x_2, y)$, but $\mathsf{Eval}(h, h(x_1), h(y)) = \mathsf{Eval}(h, h(x_2), h(y))$, since $h(x_1) = h(x_2)$. Therefore it must hold that for one of the input pairs (x_b, y) we have $\mathsf{Eval}(h, h(x_b), h(y)) \neq \text{HAMMING}_{n,t}(x_b, y)$, meaning that we get a valid attack contradicting RPPH security.

8 Randomized Robust PPH (R2P2H)

In this section, we consider a randomized notion of RPPH, denoted R2P2H, where the hash function h itself can be a randomized function. For robustness, we assume that the adversary can choose the inputs x_1, x_2 adaptively depending on the description of h, but before knowing the internal randomness that will be used in the computations of $h(x_1), h(x_2)$. The adversary wins if $\mathsf{Eval}(h, h(x_1), h(x_2)) \neq P(x_1, x_2)$. Formally, the definition is identical to that in Section 3, with two modifications:

- We now allow the hash functions $h : \{0,1\}^n \to \{0,1\}^m$ to be randomized functions.
- The definition of robustness (Definition 9) is modified accordingly so that the probability is taken also over the internal randomness used in the computation of $h(x_1), h(x_2)$.

R2P2H is a relaxation of RPPH. At first sight, it may seem that allowing randomness does not alter the problem significantly, and that RPPH and R2P2H are "morally equivalent". This is not the case. On the positive side, for interesting regimes, RPPH is known to require collision-resistant hash functions, while R2P2H can be constructed information-theoretically. On the negative side, we caution that R2P2H provides qualitatively weaker security than RPPH. For deterministic RPPH, the security definition implicitly allows the adversary to choose x_2 after seeing $h(x_1)$, since the adversary can compute $h(x_1)$ himself. This is not the case for R2P2H, where seeing $h(x_1)$ can reveal something about the internal randomness used to compute it and could allow the adversary to find a bad x_2 that breaks security. Indeed, this will be the case for our construction.

The notion of R2P2H for the equality predicate was studied implicitly in [NS96, BK97, MNS08] and the connection was recently made explicit in [CN22].

Lemma 4. *Let $T \subseteq [\ell]$ be an arbitrary set of size $|T| \geq \delta \cdot \ell$ for some constant $\delta > 0$. Let $S_A, S_B \subseteq [n]$ be chosen as uniformly random and independent sets of size $|S_A| = |S_B| = \sqrt{\lambda \ell}$ where $\ell > \lambda$. Then $\Pr[|S_A \cap S_B \cap T| = \emptyset] \leq 2^{-\Omega(\lambda)}$.*

For lack of space, we defer the proof of the above lemma to the full version.

(n, m)-R2P2H for Hamming$_{n,t}$.

Notation: For a string $y \in \{0,1\}^\ell$ and a subset $S \subseteq [\ell]$, let $y_S \in \{0,1\}^{|S|}$ denote the bits of y in the positions indexed by S.
Scheme: Set $\ell = 2n$. Let $G \in \mathbb{F}_2^{\ell \times n}$ be the generator matrix of an $[\ell, n]_2$-code which with distance $\delta\ell$ for some constant $\delta > 0$. Let $P \in \mathbb{F}_2^{(n-k)\times n}$ be a parity check matrix of an $[n, k]_2$-linear code which is efficiently list decodable against t errors.

- $h(x)$: Choose a set $S \subseteq [\ell]$ of size $|S| = \sqrt{\lambda\ell}$ uniformly at random. Let $C(x) = G \cdot x$. Output $h(x) := (P \cdot x, S, C(x)_S)$
- $\mathsf{Eval}(y_1, y_2)$: Let $y_1 = (v_1, S_1, C(x_1)_{S_1})$ and $y_2 = (v_2, S_2, C(x_2)_{S_2})$. Let $S^* = S_1 \cap S_2$. Use syndrome list-decoding for the syndrome $v_1 - v_2$ to recover a list $\mathcal{L} = \{e_1, \ldots, e_L\}$ of possible error vectors $e_i \in \mathbb{F}_2^n$ such that $Pe_i = v_1 - v_2$ and $\|e_i\|_0 \leq t$.
  - output 1, if there exists some $e_i \in \mathcal{L}$ such that $C(e_i)_{S^*} = C(x_1)_{S^*} - C(x_2)_{S^*}$.
  - otherwise output 0.

Fig. 5. Construction of (n, m)-R2P2H for Hamming$_{n,t}$

Theorem 13. *Let λ be a security parameter. For any polynomial n, t, k such that there exists an $[n, k]_2$-linear code which is efficiently list decodable against t errors, the construction above is a (n, m)-R2P2H for Hamming$_{n,t}$ with output length $m = O(\sqrt{\lambda n} \log n) + (n - k)$. In particular: (1) there exist (n, m)-RPPH*

for $\text{HAMMING}_{n,t}$ *with output length* $m = O(\sqrt{\lambda n}\log n) + 2t\log n$*, and (2) for any constant* $\eta > 0$*, there exists some constant* $\rho > 0$ *such that there exist* (n, m)*-RPPH for* $\text{HAMMING}_{n,t}$ *with* $t = \rho n$ *and* $m = O(\sqrt{\lambda n}\log n) + \eta \cdot n$*.*

Proof. Let $x_1, x_2 \in \mathbb{F}_2^n$ be arbitrary values. Let $y_1 = h(x_1)$ and $y_2 = h(x_2)$ with $y_1 = (v_1, S_1, C(x_1)_{S_1})$ and $y_2 = (v_2, S_2, C(x_2)_{S_2})$. Define T to be the set of locations where $C(e) \neq C(x_1 - x_2)$ We consider two cases:

- If $\|x_1 - x_2\|_0 \leq t$, then $e = x_1 - x_2 \in \mathcal{L}$ and $C(e)_{S^*} = C(x_1 - x_2)_{S^*} = C(x_1)_{S^*} - C(x_2)_{S^*}$. Therefore $\mathsf{Eval}(y_1, y_2)$ will output 1 with probability 1.
- If $\|x_1 - x_2\|_0 > t$, for any $e_i \in \mathcal{L}$, we have $\|e_i\|_0 \leq t$ and therefore $e_i \neq x_1 - x_2$. Define the set $T_i = \{j \ : C(x_1 - x_2)_j \neq C(e_i)_j\}$ of locations on which $C(x_1 - x_2)$ and $C(e_i)$ disagree. Since the minimum distance of the code is $\delta\ell$, we have $|T_i| > \delta\ell$. By Lemma 4, we have

  $$\Pr[\exists i \in [L] S_1 \cap S_2 \cap T_i = \emptyset] \leq \sum_{i \in [L]} \Pr[S_1 \cap S_2 \cap T_i = \emptyset] \leq L 2^{-\Omega(\lambda)} = \text{negl}(\lambda).$$

  As long as the above event does not occur, $\mathsf{Eval}(y_1, y_2)$ will output 0, since for every $i \in [L]$ we have $S_1 \cap S_2 \cap T_i \neq \emptyset$ and therefore there exists some $j \in S^*$ such that $C(x_1)_j - C(x_2)_j \neq C(e_i)_j$. Overall, this shows that $\mathsf{Eval}(y_1, y_2) = 0$ with all but negligible probability.

References

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

App. Apple csam detection. https://www.apple.com/child-safety/pdf/CSAM_Detection_Technical_Summary.pdf. Accessed: 2022-02-13.

BK97. L. Babai and P.G. Kimmel. Randomized simultaneous messages: solution of a problem of yao in communication complexity. In *Proceedings of Computational Complexity. Twelfth Annual IEEE Conference*, pages 239–246, 1997.

BLV19. Elette Boyle, Rio LaVigne, and Vinod Vaikuntanathan. Adversarially robust property-preserving hash functions. In Avrim Blum, editor, *ITCS 2019*, volume 124, pages 16:1–16:20. LIPIcs, January 2019.

BZ82. E. L Blokh and Victor Zyablov. Linear concatenated codes. *Nauka*, 1982.

CN22. Shahar P. Cohen and Moni Naor. Low communication complexity protocols, collision resistant hash functions and secret key-agreement protocols. Cryptology ePrint Archive, Paper 2022/312, 2022. https://eprint.iacr.org/2022/312.

Cru. Apple's csam detection tech is under fire — again. https://techcrunch.com/2021/08/18/apples-csam-detection-tech-is-under-fire-again/. Accessed: 2022-02-13.

DORS08. Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam D. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139, 2008.

FLS22. Nils Fleischhacker, Kasper Green Larsen, and Mark Simkin. Property-preserving hash functions from standard assumptions. *EUROCRYPT*, 2022.

FS21. Nils Fleischhacker and Mark Simkin. Robust property-preserving hash functions for hamming distance and more. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part III*, volume 12698 of *LNCS*, pages 311–337. Springer, Heidelberg, October 2021.

GHK10. Venkatesan Guruswami, Johan Håstad, and Swastik Kopparty. On the list-decodability of random linear codes. In Leonard J. Schulman, editor, *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 409–416. ACM, 2010.

GR09. Venkatesan Guruswami and Atri Rudra. Better binary list decodable codes via multilevel concatenation. *IEEE Transactions on Information Theory*, 55(1):19–26, 2009.

IM98. Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In Jeffrey Scott Vitter, editor, *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 604–613. ACM, 1998.

KOR00. Eyal Kushilevitz, Rafail Ostrovsky, and Yuval Rabani. Efficient search for approximate nearest neighbor in high dimensional spaces. *SIAM J. Comput.*, 30(2):457–474, 2000.

MNP07. Rajeev Motwani, Assaf Naor, and Rina Panigrahy. Lower bounds on locality sensitive hashing. *SIAM J. Discret. Math.*, 21(4):930–935, 2007.

MNS08. Ilya Mironov, Moni Naor, and Gil Segev. Sketching in adversarial environments. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 651–660. ACM Press, May 2008.

NS96. Ilan Newman and Mario Szegedy. Public vs. private coin flips in one round communication games (extended abstract). In *28th ACM STOC*, pages 561–570. ACM Press, May 1996.

NYT. Apple wants to protect children. but it's creating serious privacy risks. https://www.nytimes.com/2021/08/11/opinion/apple-iphones-privacy.html. Accessed: 2022-02-13.

OWZ11. Ryan O'Donnell, Yi Wu, and Yuan Zhou. Optimal lower bounds for locality sensitive hashing (except when q is tiny). In Bernard Chazelle, editor, *Innovations in Computer Science - ICS 2011, Tsinghua University, Beijing, China, January 7-9, 2011. Proceedings*, pages 275–283. Tsinghua University Press, 2011.

Ped92. Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 129–140. Springer, Heidelberg, August 1992.

Pet60. Wesley Peterson. Encoding and error-correction procedures for the bose-chaudhuri codes. *IRE Transactions on information theory*, 6(4):459–470, 1960.

RRR21. Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum. Constant-round interactive proofs for delegating computation. *SIAM J. Comput.*, 50(3), 2021.

Scha. Apple adds a backdoor to imessage and icloud storage. https://www.schneier.com/blog/archives/2021/08/apple-adds-a-backdoor-to-imesssage-and-icloud-storage.html. Accessed: 2022-02-13.

Schb. Apple's neuralhash algorithm has been reverse-engineered. https://www.schneier.com/blog/archives/2021/08/apples-neuralhash-algorithm-has-been-reverse-engineered.html. Accessed: 2022-02-13.