Robust Property-Preserving Hash Functions for Hamming Distance and More

Nils Fleischhacker[1] and Mark Simkin[2]

[1] Ruhr University Bochum, Bochum, Germany
[2] Aarhus University, Aarhus, Denmark

Abstract. Robust property-preserving hash (PPH) functions, recently introduced by Boyle, Lavigne, and Vaikuntanathan [ITCS 2019], compress large inputs x and y into short digests $h(x)$ and $h(y)$ in a manner that allows for computing a predicate P on x and y while only having access to the corresponding hash values. In contrast to locality-sensitive hash functions, a robust PPH function guarantees to correctly evaluate a predicate on $h(x)$ and $h(y)$ even if x and y are chosen adversarially *after* seeing h.
Our main result is a robust PPH function for the exact hamming distance predicate

$$\mathsf{HAM}^t(x,y) = \begin{cases} 1 & \text{if } d(x,y) \geq t \\ 0 & \text{Otherwise} \end{cases}$$

where $d(x,y)$ is the hamming-distance between x and y. Our PPH function compresses n-bit strings into $\mathcal{O}(t\lambda)$-bit digests, where λ is the security parameter. The construction is based on the q-strong bilinear discrete logarithm assumption.
Along the way, we construct a robust PPH function for the set intersection predicate

$$\mathsf{INT}^t(X,Y) = \begin{cases} 1 & \text{if } |X \cap Y| > n - t \\ 0 & \text{Otherwise} \end{cases}$$

which compresses sets X and Y of size n with elements from some arbitrary universe U into $\mathcal{O}(t\lambda)$-bit long digests. This PPH function may be of independent interest. We present an almost matching lower bound of $\Omega(t \log t)$ on the digest size of any PPH function for the intersection predicate, which indicates that our compression rate is close to optimal. Finally, we also show how to extend our PPH function for the intersection predicate to more than two inputs.

1 Introduction

Compressing data while maintaining some of its properties is one of the most fundamental tasks in computer science. Approximate set membership data structures, such as Bloom Filters [Blo70] or Cuckoo Hashing [PR04], allow for compressing large data sets into small digests that can afterwards be used to test whether some element x was a member of the original data set or not. Locality-sensitive hash functions [IM98] allow for compressing data points x and y independently into short digests $h(x)$ and $h(y)$ such that the hash values can be used to check whether the original points were close or far apart according to some metric like the euclidean or angular distance. Streaming algorithms [Mut03] enable an observer of a data stream to estimate certain statistics about the stream while using only a small amount of local storage. All of these algorithms have two things in common. They are all randomized and thus may fail on certain inputs with some, usually small, probability and they all assume that the inputs are chosen *independently* of the random coins used by the data structure.

Over the past years, a series of works [MNS08, HW13, NY15, CPS19, BLV19, BEJWY20] have investigated such data structures in the presence of adversarial inputs that are chosen *after* seeing the random coins of the data structure. Naor and Yogev [NY15], for instance, study the robustness of Bloom filters in the presence of an adversary that aims to find an input set X and a value $z \notin X$ such that the approximate membership test on a digest of X and the value z incorrectly reports that $z \in X$. Clayton, Patton, and

Shrimpton [CPS19] extend the work of Naor and Yogev to other data structures such as counting Bloom filters and count-min sketches. Boyle, LaVigne and Vaikuntanathan [BLV19], referenced as BLV hereafter, initiated the study of *robust property-preserving hash (PPH) functions*, which, in a nutshell, combine the security guarantees of collision-resistant and the functionality of locality-sensitive hash functions.

A bit more formally, a PPH function $h : X \to Y$ with evaluation algorithm $\mathsf{Eval} : Y \times Y \to \{0,1\}$ for some predicate $P : X \times X \to \{0,1\}$ is said to be robust, if no PPT adversary $\mathcal{A}$, who is given (h, Eval), can produce an output (x, y) such that $P(x, y) \neq \mathsf{Eval}(h(x), h(y))$. The authors construct such a hash function, which compresses n-bit inputs by some small constant factor, for the gap-hamming predicate

$$\mathsf{GAP\text{-}HAM}^t_\epsilon(x,y) = \begin{cases} 1 & \text{if } d(x,y) \geq t(1+\epsilon) \\ 0 & \text{if } d(x,y) \leq t(1-\epsilon) \\ \star & \text{Otherwise} \end{cases}$$

for $t = \mathcal{O}(n/\log n)$ and an arbitrary small, but constant, non-zero value ϵ, where d is the hamming distance. This means that the Eval function can only guarantee that the Hamming distance is *at most* $t(1+\epsilon)$ for output 0 or *at least* $t(1-\epsilon)$ for output 1. In the gap between the two values either output is possible and we get no correctness or security guarantees. It would therefore be desirable to close this gap, i.e., to obtain a construction for $\epsilon = 0$, which was left open by the work of BLV.

1.1 Our Contribution

In this work, we construct a robust PPH function for the exact hamming distance predicate, which essentially corresponds to $\mathsf{GAP\text{-}HAM}^t_\epsilon(x,y)$ with $\epsilon = 0$, for $t \leq n/c\lambda$ for some small constant $c > 1$. We also show how to generalize our result to strings over large alphabets, e.g. alphanumeric sequences, and the corresponding generalized hamming distance, which counts the number of positions in which the strings differ.[3] Our construction is based on the q-strong bilinear discrete logarithm assumption in pairing-friendly groups and compresses n-bit inputs into $\mathcal{O}(t\lambda)$-bit hash values.

Along the way, we consider the symmetric set difference predicate, which takes two sets X and Y of size n from some universe U as input and checks whether $|(X \setminus Y) \cup (Y \setminus X)| < t$. We construct a PPH function for this predicate from the same assumptions and with the same $\mathcal{O}(t\lambda)$-bit long hash values as above. Here it is insightful to note that for two-input predicates, the symmetric set difference and an intersection predicate with a threshold on the minimum intersection size are equivalent.

For the symmetric set difference and the intersection predicate, we show that any PPH function has to have $\Omega(t \log t)$-bit long hash values, which indicates that our hash functions are close to optimal in terms of compression factor.

Finally, we show how to construct PPH functions for the intersection predicate with more than two inputs.

1.2 Technical Overview

We will start our overview by constructing a robust PPH function for the two-input symmetric set difference predicate. Obtaining our PPH function for the exact hamming distance predicate will only require one additional step of encoding the input bit strings into appropriate sets.

The starting point of our work is a simple, yet beautiful, observation about polynomials and rational function interpolation made by Minsky et al. [MTZ03][4]. Consider sets $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_n\}$

[3] Note that encoding strings from a large alphabet into bit strings and then using our construction for binary inputs does not work, since the hamming distance of the encoded strings has no meaningful interpretation.

[4] The work of Minsky et al. has recently found other applications in the context of cryptography in the domain of communication efficient private set intersection protocols [GS19a].

which are encoded into the roots of some polynomials $u(x) = \prod_{i=0}^{n}(x - a_i)$ and $v(x) = \prod_{i=0}^{n}(x - b_i)$ over some finite field $\mathbb{F}$, and consider the rational function[5]

$$w(x) := \frac{u(x)}{v(x)} = \frac{\prod_{a_i \notin A \setminus B}(x - a_i)}{\prod_{b_j \notin B \setminus A}(x - b_j)}.$$

The main observation behind the work of Minsky et al. was the following: the larger the intersection of A and B, the smaller their symmetric set difference, the more roots of the polynomials $u(x)$ and $v(x)$ "cancel out". Furthermore, the smaller the degrees of the remaining polynomials in the numerator and denominator, the fewer evaluation points are needed for correctly interpolating $w(x)$. More precisely, the degree each polynomial, once they have been reduced to lowest terms, is exactly $|A \setminus B| = |B \setminus A|$, thus if symmetric set difference is at most $2t$ large, then $w(x)$ can be correctly interpolated from ℓ evaluation points of $w(x)$, where $\ell \in \mathcal{O}(t)$. Importantly for us, ℓ can be chosen such that $\ell - 1$ points are not sufficient for correctly interpolating the rational function if $|(A \setminus B) \cup (B \setminus A)| > 2t$.

As a first attempt towards compressing sets A and B of size n into appropriate hash values, one might want to compute $(u(\alpha_1), \ldots, u(\alpha_\ell))$ and $(v(\alpha_1), \ldots, v(\alpha_\ell))$, where $\alpha_1, \ldots, \alpha_\ell$ are some distinct publicly known fixed evaluation points. Given these two hash values, the evaluation algorithm Eval could compute $w(\alpha_i) := u(\alpha_i)/v(\alpha_i)$ for $i \in [n]$ and attempt to interpolate a rational function $\hat{w}(x)$ using these points. Recall that $w(x) = \hat{w}(x)$ if $|(A \setminus B) \cup (B \setminus A)| \leq 2t$ and $w(x) \neq \hat{w}(x)$ otherwise.

At this point we are left with the task of checking whether the interpolated function is the correct one. Ideally, we would like to simply evaluate the polynomials $u(x)$ and $v(x)$ on some random point r and check whether $u(r)/v(r) = \hat{w}(r)$. Over a large enough field and using a uniformly and independently sampled random value r this allows us to efficiently test the equality of two (rational) functions with only negligible error. Unfortunately, since we are designing a hash function, rather than an interactive protocol, $u(r)$ and $v(r)$ would need to be part of the corresponding hash value. This means, r would need to be fixed at the time of hashing and needs to be the same for all inputs to the hash function. Thus, it has to already be fixed as part of the sampling of the hash function from its corresponding family and an adversary can choose sets A and B *conditioned* on r. Since r is now no longer distributed independently of A, B the adversary could potentially find two such input sets with $|(A \setminus B) \cup (B \setminus A)| > 2t$, which result in an interpolation of a function $\hat{w}(x) \neq w(x)$ that still passes the check, because the sets are chosen such that $\hat{w}(r) = w(r)$.

To get around this problem, we need to hide r from the adversary. Towards this goal, we fix a uniformly random *hidden* value r in a way that allows for performing the check described above *obliviously*. Assume the PPH function description includes values $\boldsymbol{\Gamma} = (g, g^r, g^{r^2}, \ldots, g^{r^n})$ for some uniformly random value r. Now given the coefficients of polynomials $u(x)$, $v(x)$, and $\boldsymbol{\Gamma}$, we can evaluate our polynomials in the exponent to obtain $g^{u(r)}$ and $g^{v(r)}$. Under an appropriate q-type discrete logarithm assumption we can argue that the actual value r remains hidden and the attack outlined above is no longer possible.

To see how to perform the rational function equality check in the exponent, assume that the interpolation of $\hat{w}(x)$ gives us the coefficients of the polynomials $\hat{v}(x)$ and $\hat{u}(x)$ with $\hat{w}(x) = \hat{u}(x)/\hat{v}(x)$. The equation

$$w(x) = \frac{u(x)}{v(x)} = \frac{\hat{u}(x)}{\hat{v}(x)} = \hat{w}(x)$$

holds if and only if

$$u(x)\hat{v}(x) = \hat{u}(x)v(x)$$

holds. Finally, given $g^{u(r)}, g^{v(r)}$, which are computed independently during hashing, the vector $\boldsymbol{\Gamma}$, which is part of the hash function description, and the coefficients of $\hat{u}(x)$ and $\hat{v}(x)$, which we obtain from the interpolation, we can use a bilinear pairing, which allows us to perform a multiplication in the exponent, to check the desired equation.

[5] Note, that the equality does not strictly hold, since the function on the right is defined for $x \in A \cup B$, whereas the one on the left is not. However, the two functions are equivalent for all x except for the removable singularities of $u(x)/v(x)$ which is exactly what we need.

To obtain our construction for the hamming distance predicate, we need to encode the input bit strings into sets in a way that allows us to translate a threshold on the hamming distance to a threshold on the size of the symmetric set difference of the corresponding sets. Towards this goal, we simply encode a bit string $x = x_1 x_2 \dots x_n$ into a set $S_x \coloneqq \{2i - x_i \mid i \in [n]\}$. For two strings x and y and each bit position i with $x_i = y_i$ the corresponding sets S_x and S_y will have one element $2i - x_i = 2i - y_i$ in common. For each position i with $x_i \neq y_i$, the sets will contain distinct elements $2i$ and $2i - 1$. With this in mind, it is straightforward to see that

$$d(x, y) = \frac{|(S_x \setminus S_y) \cup (S_y \setminus S_x)|}{2},$$

which means that we can reduce the problem of computing the hamming distance between bit strings to computing the size of the symmetric set difference of the corresponding set encodings.

2 Preliminaries

This section introduces notation, some basic definitions and lemmas that we will use throughout this work. We denote by $\lambda \in \mathbb{N}$ the security parameter and by $\mathsf{poly}(\lambda)$ any function that is bounded by a polynomial in λ. A function f is negligible if for every $c \in \mathbb{N}$, there exists some $N \in \mathbb{N}$ such that for all $\lambda > N$ it holds that $f(\lambda) < 1/\lambda^c$. We denote by $\mathsf{negl}(\lambda)$ any negligible function. An algorithm is PPT if it is modeled by a probabilistic Turing machine with a running time bounded by $\mathsf{poly}(\lambda)$.

Let $n \in \mathbb{N}$, we denote by $[n]$ the set $\{1, \dots, n\}$. Let X, Y be sets, we denote by $|X|$ the size of X and by $X \triangle Y$ the symmetric set difference of X and Y, i.e., $X \triangle Y = (X \cup Y) \setminus (X \cap Y) = (X \setminus Y) \cup (Y \setminus X)$. Further, we denote by $\mathcal{P}_n(X) = \{S \subseteq X \mid |S| = n\}$ the set of all subsets of size n of X and by $x \leftarrow X$ the process of sampling an element of X uniformly at random. Let $x, y \in \{0,1\}^n$, we write $w(x)$ to denote the Hamming weight of x and we write $d(x, y)$ to denote the Hamming distance between x and y, i.e., $d(x, y) = w(x \oplus y)$. For a polynomial $p = \sum_{i=0}^{n} c_i x^i$, we write $\mathsf{coef}(p, i) = c_i$ to denote the i-th coefficient of p.

Rational Functions. A rational function is the fraction of two polynomials. The total degree of a rational function is the sum of the degrees of the numerator and the denominator after they have been reduced to lowest terms. More precisely, it is defined as follows.

Definition 1 (Total Degree). *Let f and g be arbitrary non-zero polynomials. Let r, f', g' be polynomials, such that $f = rf'$, $g = rg'$ and f' and g' are co-prime. Note that r, f', g' always exist and are unique. The total degree of the rational function f/g is then defined as $\mathsf{tdeg}(f/g) = \mathsf{deg}(f') + \mathsf{deg}(g')$.*

Encoding bit strings as sets. A given bit string $x \in \{0,1\}^n$ can be efficiently encoded into a set as $S_x \coloneqq \{2i - x_i \mid i \in [n]\}$. We have that $S_x \in \mathcal{P}_n([2n])$, i.e. the size of S_x is n and its description length in bits is $n\lceil \log 2n \rceil$. We call S_x the *set encoding* of x.

Lemma 1. *Let $n \in \mathbb{N}$. For any $x, y \in \{0,1\}^n$, it holds that*

$$2d(x, y) = |S_x \triangle S_y|.$$

Proof. We denote by $I \coloneqq \{i \in [n] \mid x_i = y_i\}$ the set of indices i where $x_i = y_i$. Similarly, we denote by $J \coloneqq \{j \in [n] \mid x_j \neq y_j\}$ the set of indices j where $x_j \neq y_j$. By definition of the Hamming distance, we have $|J| = d(x, y)$ and $|I| = n - |J|$.

We can now write S_x, S_y in terms of I and J as

$$\begin{aligned} S_x &= \{2i - x_i \mid i \in I\} \cup \{2j - x_j \mid j \in J\} \\ S_y &= \{2i - y_i \mid i \in I\} \cup \{2j - y_j \mid j \in J\}. \end{aligned}$$

By definition of I, we have that $\{2i - x_i \mid i \in I\} = \{2i - y_i \mid i \in I\}$ and therefore that

$$S_x \cup S_y = \{2i - x_i \mid i \in [n]\} \cup \{2j - y_j \mid j \in J\} = S_x \cup \{2j - y_j \mid j \in J\}$$

Since, by definition of J it must also hold that $S_x \cap \{2j - y_j \mid j \in J\} = \emptyset$ we thus have

$$|S_x \cup S_y| = |S_x| + |J| = n + d(x, y). \tag{1}$$

Similarly, by the above observations, it holds that $S_x \cap S_y = \{2i - x_i \mid i \in I\}$ and thereby

$$|S_x \cap S_y| = |I| = n - |J| = n - d(x, y) \tag{2}$$

Finally, combining the definition of symmetric set difference and Equations 1 and 2 we thus have

$$\begin{aligned} |S_x \triangle S_y| &= |(S_x \cup S_y) \setminus (S_x \cap S_y)| = |S_x \cup S_y| - |S_x \cap S_y| \\ &= n + d(x, y) - (n - d(x, y)) = 2d(x, y) \end{aligned}$$

as claimed. □

Encoding sets as polynomials. We define the *polynomial encoding* of a set $S = \{s_1, \ldots, s_n\} \subseteq [N]$ as the polynomial $p_S(z) = \prod_{i=1}^{n}(z - s_i)$ over some field $\mathbb{Z}_q$ of prime order $q > N$. For a bit string $x \in \{0,1\}^n$, we will abuse notation and write p_x to denote the polynomial encoding of the set encoding of x. Observe that the roots of p_x are all in $[2n]$.

Lemma 2. *Let $n, N \in \mathbb{N}$ such that $n < N$. For any pair of sets $X, Y \in \mathcal{P}_n([N])$, it holds that*

$$|X \triangle Y| = \mathsf{tdeg}\left(\frac{p_X}{p_Y}\right).$$

Proof. Let $X' \coloneqq X \setminus Y$, $Y' \coloneqq Y \setminus X$ and $W \coloneqq X \cap Y$. We have by definition of the polynomial encoding that

$$p_X(z) = \prod_{x \in X}(z - x) = \left(\prod_{w \in W}(z - w)\right) \cdot \left(\prod_{x \in X'}(z - x)\right)$$

and

$$p_Y(z) = \prod_{y \in Y}(z - y) = \left(\prod_{w \in W}(z - w)\right) \cdot \left(\prod_{y \in Y'}(z - y)\right).$$

Since $X' \cap Y' = \emptyset$, the two polynomials $\prod_{x \in X'}(z - x)$ and $\prod_{y \in Y'}(z - y)$ are coprime, while $\prod_{w \in W}(z - w)$ is a common factor in p_X and p_Y. By Definition 1, it thus holds that

$$\begin{aligned} \mathsf{tdeg}\left(\frac{p_X(z)}{p_Y(z)}\right) &= \mathsf{deg}\left(\prod_{x \in X'}(z - x)\right) + \mathsf{deg}\left(\prod_{y \in Y'}(z - y)\right) \\ &= |X'| + |Y'| = |X' \cup Y'| = |(X \setminus Y) \cup (Y \setminus X)| \\ &= |X \triangle Y| \end{aligned}$$

as claimed. □

Proposition 3 ([MTZ03]). *For polynomials $f \in \mathbb{F}_{\leq n}[X]$ and $g \in \mathbb{F}_{\leq m}[X]$, the rational function $h(z) = f(z)/g(z)$ can be uniquely interpolated (up to equivalences) from distinct evaluation points $z_1, \ldots, z_d$ and $f(z_1), g(z_1), \ldots, f(z_d), g(z_d)$, where $d = n + m + 1$, as well as upper bounds on n and m.*

Remark 1. We denote by RatInt the algorithm that takes as input a list of d points $(x_1, y_1), \ldots, (x_d, y_d) \in \mathbb{F}_q$ and tries to find a rational function p/q with degrees of p and q at most $\lfloor (d-1)/2 \rfloor$, such that $p(x_i)/q(x_i) = y_i$ for $1 \leq i \leq d$. Upon success it outputs p/q. Otherwise it outputs the constant 0 function.

Two-Input Predicates. We define the following two-input predicates, which will be the main focus of this work.

Definition 2 (Hamming Predicate). *For $x, y \in \{0,1\}^n$ and $t > 0$, the two-input predicate is defined as*

$$\mathsf{HAM}^t(x,y) = \begin{cases} 1 & \text{if } d(x,y) \geq t \\ 0 & \text{Otherwise} \end{cases}$$

Definition 3 (Symmetric Set Difference Predicate). *For a universe U, natural number n, $X, Y \in \mathcal{P}_n(U)$, and $t > 0$, the two-input symmetric set difference predicate is defined as*

$$\mathsf{SSD}^t(X,Y) = \begin{cases} 1 & \text{if } |X \triangle Y| \geq t \\ 0 & \text{Otherwise} \end{cases}$$

Bilinear Groups and Pairings. A bilinear group is described by a tuple $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, e)$, where $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are groups of order q and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a (non-degenerate) bilinear asymmetric map, called pairing, such that for all $a, b \in \mathbb{Z}_q$ and $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$ it holds that

$$e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}.$$

If $\mathbb{G}_1$ and $\mathbb{G}_2$ are cyclic and g_1 and g_2 are generators of those groups respectively, then $e(g_1, g_2)$ is a generator of $\mathbb{G}_T$.

Let GGen be a PPT algorithm that take as input the security parameter 1^λ and outputs bilinear map parameters $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, e, g_1, g_2)$, where $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are the groups of prime order $q = q(\lambda)$. $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a (non-degenerate) bilinear map and g_1 and g_2 are generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively. Our constructions will rely on the following q-type extension of the discrete logarithm assumption over bilinear groups.

Definition 4 (q-Strong Bilinear Discrete Logarithm (q-SBDL) Assumption). *The q-sBDL assumption holds relative to GGen if for all PPT algorithms $\mathcal{A}$ it holds that*

$$\Pr\left[r = \mathcal{A}\left(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, e, \begin{pmatrix} g_1 & g_1^r & \cdots & g_1^{r^n} \\ g_2 & g_2^r & \cdots & g_2^{r^n} \end{pmatrix}\right)\right] \leq \mathsf{negl}(\lambda)\,,$$

where the probability is taken over $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, e, g_1, g_2) \leftarrow \mathsf{GGen}(1^\lambda)$ and $r \leftarrow \mathbb{Z}_q$.

To the best of our knowledge, and unlike the regular q-strong discrete logarithm (q-SDL) assumption [GOR11], this exact assumption has not been used before. However, it is in fact implied by, and thus weaker than, other related q-type assumptions such as the q-BDHI [BB04] assumption.

Hash Functions. We first recall the standard definition of collision-resistant hash functions.

Definition 5 (Collision Resistant Hash Function Family). *For a $\lambda \in \mathbb{N}$ a hash function family $\mathcal{F} = \{f : \{0,1\}^* \to \{0,1\}^\lambda\}$ consists of a pair of efficiently computable algorithms:*

$\mathsf{Sample}(1^\lambda) \to f$ *is an efficient randomized algorithm that samples an efficiently computable random hash function from $\mathcal{F}$ with security parameter λ.*

$\mathsf{Hash}(f, x) \to y$ *is an efficient deterministic algorithm that evaluates the hash function h on x.*

The family $\mathcal{F}$ is collision resistant if, for any PPT adversary $\mathcal{A}$ it holds that,

$$\Pr[f \leftarrow \mathsf{Sample}(1^\lambda); (x_1, x_2) \leftarrow \mathcal{A}(f) : f(x_1) = f(x_2)] \leq \mathsf{negl}(\lambda)\,,$$

where the probability is taken over the internal random coins of Sample and $\mathcal{A}$.

The following definition of property-preserving hash functions is taken almost verbatim from [BLV19]. In this work, we consider the strongest of several different security notions that were proposed by BLV.

Definition 6 (Property-Preserving Hash). *For a $\lambda \in \mathbb{N}$ an η-compressing property preserving hash function family $\mathcal{H}_\lambda = \{h : X \to Y\}$ for a two-input predicate $P : X \times X \to \{0,1\}$ requires the following three efficiently computable algorithms:*

$\mathsf{Sample}(1^\lambda) \to h$ *is an efficient randomized algorithm that samples an efficiently computable random hash function from $\mathcal{H}$ with security parameter λ.*

$\mathsf{Hash}(h, x) \to y$ *is an efficient deterministic algorithm that evaluates the hash function h on x.*

$\mathsf{Eval}(h, y_1, y_2) \to \{0,1\}$**:** *is an efficient deterministic algorithm that on input h, and $y_1, y_2 \in Y$ outputs a single bit.*

We require that $\mathcal{H}$ must be compressing, meaning that $\log |Y| \leq \eta \log |X|$ for $0 < \eta < 1$.

For notational convenience we write $h(x)$ for $\mathsf{Hash}(h, x)$.

Definition 7 (Direct-Access Robustness). *A family of PPH functions $\mathcal{H} = \{h : X \to Y\}$ for a two-input predicate $P : X \times X \to \{0,1\}$ is a family of direct-access robust PPH functions if, for any PPT adversary $\mathcal{A}$ it holds that,*

$$\Pr\left[\begin{matrix} h \leftarrow \mathsf{Sample}(1^\lambda); \\ (x_1, x_2) \leftarrow \mathcal{A}(h) \end{matrix} : \mathsf{Eval}(h, h(x_1), h(x_2)) \neq P(x_1, x_2)\right] \leq \mathsf{negl}(\lambda)\,,$$

where the probability is taken over the internal random coins of Sample *and $\mathcal{A}$.*

3 PPH for Symmetric Set Difference

In this section we construct property preserving hash functions for symmetric set difference. We start by presenting a construction for sets with elements from a universe of bounded size in Section 3.1 and show how to extend the construction to sets with elements from an arbitrarily large universe in Section 3.2.

3.1 PPH for Symmetric Set Difference of $\mathcal{P}_n([N])$

Theorem 4. *Let* GGen *be a bilinear group generation algorithm that generates groups of prime order $q = q(\lambda)$ with $q > 2^\lambda$. Then, for any $n = \mathsf{poly}(\lambda)$, $N \in \mathbb{N}$ with $n \leq N \leq 2^{\lambda-1}$ and any $t < \frac{n(\log N - \log n)}{\log q(\lambda)} - 1$, the construction in Figure 1 is a $\frac{(t+1)\log q(\lambda)}{\log \binom{N}{n}} \leq \frac{(t+1)\log q(\lambda)}{n(\log N - \log n)}$-compressing direct-access robust property preserving hash function family for the two-input predicate SSD^t and domain $\mathcal{P}_n([N])$, if the n-SBDL assumption holds relative to* GGen.

Proof. Let $\mathcal{A}$ be an arbitrary PPT adversary against the direct access robustness of $\mathcal{H}$. We have

$$\begin{aligned} & \Pr[\mathsf{Eval}(h, h(X_1), h(X_2)) \neq \mathsf{SSD}^t(X_1, X_2)] \\ = \; & \Pr[\mathsf{Eval}(h, h(X_1), h(X_2)) = 1 \mid \mathsf{SSD}^t(X_1, X_2) = 0] \cdot \Pr[\mathsf{SSD}^t(X_1, X_2) = 0] \\ & + \Pr[\mathsf{Eval}(h, h(X_1), h(X_2)) = 0 \mid \mathsf{SSD}^t(X_1, X_2) = 1] \cdot \Pr[\mathsf{SSD}^t(X_1, X_2) = 1], \end{aligned}$$

where the probabilities are taken over $h \leftarrow \mathsf{Sample}(1^\lambda)$ and $(X_1, X_2) \leftarrow \mathcal{A}(h)$. We consider the two cases separately.

Claim 5. $\Pr[\mathsf{Eval}(h, h(X_1), h(X_2)) = 1 \mid \mathsf{SSD}^t(X_1, X_2) = 0] = 0$

$\mathsf{Sample}(1^\lambda)$

$(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, e, g_1, g_2) \leftarrow \mathsf{GGen}(1^\lambda)$

$r \leftarrow \mathbb{Z}_q \setminus [N]$

$\boldsymbol{\Gamma} \coloneqq \begin{pmatrix} g_1 & g_1^r & \cdots & g_1^{r^n} \\ g_2 & g_2^r & \cdots & g_2^{r^n} \end{pmatrix}$

return $h \coloneqq (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, e, \boldsymbol{\Gamma})$

$\mathsf{Hash}(h, X)$

parse h **as** $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, e, \boldsymbol{\Gamma})$

$\boldsymbol{a} \coloneqq \big(p_X(N+1), \ldots, p_X(N+t)\big)$

$b \coloneqq \prod_{i=0}^{n} \Gamma_{1,i+1}^{\mathsf{coef}(p_X, i)} = g_1^{p_X(r)}$

return $(\boldsymbol{a}, b)$

$\mathsf{Eval}(h, (\boldsymbol{a}, b), (\tilde{\boldsymbol{a}}, \tilde{b}))$

parse h **as** $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, e, \boldsymbol{\Gamma})$

for $1 \le i \le t$

$s_i \coloneqq \left(N + i, \frac{a_i}{\widetilde{a}_i}\right)$

$(u, v) \coloneqq \mathsf{RatInt}(s_1, \ldots, s_t)$

return $e\left(b, \prod_{i=0}^{n} \Gamma_{2,i+1}^{\mathsf{coef}(v,i)}\right) \stackrel{?}{\neq} e\left(\tilde{b}, \prod_{i=0}^{n} \Gamma_{2,i}^{\mathsf{coef}(u,i)}\right)$

Fig. 1. A family of direct-access robust PPH for the predicate SSD^t over the domain $\mathcal{P}_n([N])$ for any $N \in \mathbb{N}$ with $N \le 2^{\lambda-1}$.

Proof (Claim 5). Since $\mathsf{SSD}^t(X_1, X_2) = 0$, we know that $|X_1 \triangle X_2| < t$. By Lemma 2 this means that

$$\mathsf{tdeg}\left(\frac{p_{X_1}}{p_{X_2}}\right) < t.$$

From Proposition 3 it follows that the rational function p_{X_1}/p_{X_2} can (up to equivalences) be uniquely interpolated from t points. We observe that for $1 \le i \le t$ it holds that $p_{X_2}(N+i) \neq 0$, since roots of p_{X_2} are in the interval $[N]$ by construction. Therefore, $s_i = p_{X_1}(N+i)/p_{X_2}(N+i)$ is well-defined and thus

$$\frac{p_{X_1}}{p_{X_2}} = \frac{u}{v}$$

where u/v is the rational function computed by RatInt in $\mathsf{Eval}(h, h(X_1), h(X_2))$. Finally, we observe that

$$\begin{aligned} & e\left(g_1^{p_{X_1}(r)}, \prod_{i=0}^{n} \Gamma_{2,i+1}^{\mathsf{coef}(v,i)}\right) = e\left(g_1^{p_{X_2}(r)}, \prod_{i=0}^{n} \Gamma_{2,i}^{\mathsf{coef}(u,i)}\right) \\ \iff & e(g_1, g_2)^{p_{X_1}(r) \sum_{i=0}^{n} \mathsf{coef}(v,i) \cdot r^i} = e(g_1, g_2)^{p_{X_2}(r) \sum_{i=0}^{n} \mathsf{coef}(u,i) \cdot r^i} \\ \iff & e(g_1, g_2)^{p_{X_1}(r) v(r)} = e(g_1, g_2)^{p_{X_2}(r) u(r)}, \end{aligned}$$

which is true whenever

$$\begin{aligned} & p_{X_1}(r) \cdot v(r) = p_{X_2}(r) \cdot u(r) \\ \iff & \frac{p_{X_1}(r)}{p_{X_2}(r)} = \frac{u(r)}{v(r)}, \end{aligned}$$

which is true for all r and thus the last inequality in $\mathsf{Eval}(h, h(X_1), h(X_2))$ is never satisfied. □

Claim 6. *If the n-SBDL assumption holds relative to* GGen*, then*

$$\begin{aligned}&\Pr[\mathsf{Eval}(h, h(X_1), h(X_2)) = 0 \mid \mathsf{SSD}^t(X_1, X_2) = 1] \cdot \Pr[\mathsf{SSD}^t(X_1, X_2) = 1] \\ \leq &\mathsf{negl}(\lambda)\,.\end{aligned}$$

Proof (Claim 6). Since $\mathsf{SSD}^t(X_1, X_2) = 1$, it must hold that $t \leq |X_1 \triangle X_2| \leq 2n$. By Lemma 2 this means that

$$t \leq \mathsf{tdeg}\left(\frac{p_{x_1}}{p_{x_2}}\right) \leq 2n.$$

On the other hand, by construction u/v is the rational function of total degree at most $t-1$ uniquely determined by $s_1, \ldots, s_t$. It must therefore hold that

$$\frac{u}{v} \neq \frac{p_{x_1}}{p_{x_2}}.$$

For the last inequality in $\mathsf{Eval}(h, h(X_1), h(X_2))$ to hold, p_{X_1}/p_{X_2} and u/v must therefore be two *different* rational functions that agree on point r. This means that r must be one of the at most $n + (t-1)/2$ roots of the rational function

$$\frac{p_{X_1}}{p_{X_2}} - \frac{u}{v} = \frac{p_{X_1} \cdot v - p_{X_2} \cdot u}{p_{X_2} \cdot v}.$$

Whenever $\mathcal{A}$ would be successful, we could therefore find r by testing the roots of the polynomial $p_{X_1} \cdot v - p_{X_2} \cdot u$. We give a formal reduction as follows:

$\mathcal{R}$ takes as input

$$\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, e, \boldsymbol{\Gamma} \coloneqq \begin{pmatrix} g_1 & g_1^r & \cdots & g_1^{r^n} \\ g_2 & g_2^r & \cdots & g_2^{r^n} \end{pmatrix}$$

and invokes $\mathcal{A}$ on $h \coloneqq (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, e, \boldsymbol{\Gamma})$ and receives X_1, X_2.

If $\mathsf{SSD}^t(X_1, X_2) = 0$, $\mathcal{R}$ aborts. Otherwise it computes (u, v) as in Eval and determines the set X of roots of the polynomial $p_{X_1} \cdot v - p_{X_2} \cdot u$. For each $r' \in X$, $\mathcal{R}$ checks whether $g_1^{r'} \stackrel{?}{=} \Gamma_{1,2}$ and returns r' if it holds. If the equality holds for no $r' \in X$, $\mathcal{R}$ aborts.

Since $\mathcal{A}$ is PPT and finding the roots of a polynomial is possible in polynomial time, it follows that $\mathcal{R}$ is PPT and must, by assumption, have a negligible success probability against the n-SBDL problem.

Note that r from the input of the reduction is distributed uniformly in $\mathbb{Z}_q$, while $\mathcal{A}$ expects r to be uniformly distributed in $\mathbb{Z}_q \setminus [N]$. However, since $N \leq 2^{\lambda-1}$ and $q > 2^\lambda$, it holds that $r \in \mathbb{Z}_q \setminus [N]$ with probability at least $1/2$. Furthermore, once we condition on $r \notin [N]$, the distribution of h is identical to the one expected by $\mathcal{A}$.

Now, observe that the reduction $\mathcal{R}$ is successful, if $\mathcal{A}$ outputs X_1, X_2, such that $\mathsf{SSD}^t(X_1, X_2) = 1$ and r is one of the roots of $p_{X_1} \cdot v - p_{X_2} \cdot u$. As argued above, the latter must be true, if $\mathsf{Eval}(h, h(x_1), h(x_2)) = 0$. Therefore, it holds that

$$\begin{aligned}&\mathsf{negl}(\lambda) \\ \geq &\Pr\left[r = \mathcal{R}\left(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, e, \begin{pmatrix} g_1 & g_1^r & \cdots & g_1^{r^n} \\ g_2 & g_2^r & \cdots & g_2^{r^n} \end{pmatrix}\right)\right] \\ \geq &\Pr\left[r \notin [N]\right] \cdot \Pr\left[r = \mathcal{R}\left(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, e, \begin{pmatrix} g_1 & g_1^r & \cdots & g_1^{r^n} \\ g_2 & g_2^r & \cdots & g_2^{r^n} \end{pmatrix}\right) \;\middle|\; r \notin [N]\right] \\ \geq &\frac{1}{2} \cdot \Pr[\mathsf{Eval}(h, h(X_1), h(X_2)) = 0 \mid \mathsf{SSD}^t(X_1, X_2) = 1] \cdot \Pr[\mathsf{SSD}^t(X_1, X_2) = 1]\end{aligned}$$

and, thus, the claim follows. □

Using Claims 5 and 6 we can thus conclude that

$$\Pr[\mathsf{Eval}(h, h(X_1), h(X_2)) \neq \mathsf{SSD}^t(X_1, X_2)] \leq 0 + \mathsf{negl}(\lambda) = \mathsf{negl}(\lambda)\,.$$

Therefore, $\mathcal{H}$ is direct access robust as claimed. It remains to show that it is also compressing. The domain of the hash function is $\mathcal{P}_n([N])$, the codomain is $\mathbb{Z}_{q(\lambda)}^t \times \mathbb{G}_1$. It follows that the compression factor is

$$\eta = \frac{\log |\mathbb{Z}_{q(\lambda)}^t \times \mathbb{G}_1|}{\log |\mathcal{P}_n([N])|} = \frac{\log q(\lambda)^{t+1}}{\log \binom{N}{n}} \leq \frac{\log q(\lambda)^{t+1}}{\log \left(\frac{N}{n}\right)^n} = \frac{(t+1)\log q(\lambda)}{n(\log N - \log n)}$$

as claimed. The construction is thus compressing, if

$$\frac{(t+1)\log q(\lambda)}{n(\log N - \log n)} < 1 \iff t < \frac{n(\log N - \log n)}{\log q} - 1.$$

□

3.2 PPH for Symmetric Set Difference of Arbitrary Sets.

To obtain our construction for sets with elements from an arbitrarily large universe, we make use of a collision-resistant hash function. We simply first hash the elements of the input sets into a smaller universe and then apply our construction from the previous section.

$\mathsf{Sample}(1^\lambda)$	$\mathsf{Hash}((h,f),X)$	$\mathsf{Eval}((h,f),y,\tilde{y})$
$h \leftarrow \mathcal{H}.\mathsf{Sample}(1^\lambda)$	$X' \coloneqq \{f(x) \mid x \in X\}$	$b \coloneqq \mathcal{H}.\mathsf{Eval}(h,y,\tilde{y})$
$f \leftarrow \mathcal{F}.\mathsf{Sample}(1^\lambda)$	$y \coloneqq h(X')$	**return** b
return $h' \coloneqq (h,f)$	**return** y	

Fig. 2. A family of direct-access robust PPH for SSD^t on $\mathcal{P}_n(\{0,1\}^\ell)$.

Theorem 7. *Let $\mathcal{H}_\lambda = \{h : \mathcal{P}_n(\{0,1\}^\lambda) \to Y\}$ be an η-compressing direct-access robust property preserving hash function family for SSD^t. Let $\mathcal{F} = \{f : \{0,1\}^\ell \to \{0,1\}^\lambda\}$ be a collision resistant hash function family. Then the construction in Figure 2 is a $\eta \cdot \frac{\log \binom{2^\lambda}{n}}{\log \binom{2^\ell}{n}} \leq \eta \cdot \frac{\log e + \lambda - \log n}{\ell - \log n}$-compressing direct-access robust PPH for SSD^t and domain $\mathcal{P}_n(\{0,1\}^\ell)$.*

Proof. Let $\mathcal{A}$ be an arbitrary PPT adversary against the direct-access robustness of $\mathcal{H}'$. We have that

$$\begin{aligned}
&\Pr[\mathsf{Eval}(h', h'(X_1), h'(X_2)) \neq \mathsf{SSD}^t(X_1, X_2)] \\
&= \Pr[\mathcal{H}.\mathsf{Eval}(h, h(X_1'), h(X_2')) \neq \mathsf{SSD}^t(X_1, X_2)] \\
&= \Pr\Big[\mathcal{H}.\mathsf{Eval}(h, h(X_1'), h(X_2')) \neq \mathsf{SSD}^t(X_1, X_2) \;\Big|\; |X_1' \triangle X_2'| = |X_1 \triangle X_2|\Big] \\
&\quad \cdot \Pr\Big[|X_1' \triangle X_2'| = |X_1 \triangle X_2|\Big] \\
&+ \Pr\Big[\mathcal{H}.\mathsf{Eval}(h, h(X_1'), h(X_2')) \neq \mathsf{SSD}^t(X_1, X_2) \;\Big|\; |X_1' \triangle X_2'| \neq |X_1 \triangle X_2|\Big] \\
&\quad \cdot \Pr\Big[|X_1' \triangle X_2'| \neq |X_1 \triangle X_2|\Big]
\end{aligned} \tag{3}$$

where the probability is taken over the sampling of $h' = (h,f) \leftarrow \mathsf{Sample}'(1^\lambda)$ and $(X_1, X_2) \leftarrow \mathcal{A}(h')$. Equation 3 follows by applying the definition of $\mathcal{H}'$ and then splitting the probability. We will now upper bound the two parts of the sum in Claims 8 and 9.

Claim 8. *If $\mathcal{H}$ is direct-access robust, it holds that*

$$\Pr\Big[\mathcal{H}.\mathsf{Eval}(h, h(X_1'), h(X_2')) \neq \mathsf{SSD}^t(X_1, X_2) \;\Big|\; |X_1' \triangle X_2'| = |X_1 \triangle X_2|\Big] \cdot \Pr\Big[|X_1' \triangle X_2'| = |X_1 \triangle X_2|\Big] \leq \mathsf{negl}(\lambda)\,.$$

Proof (Claim 8). By the direct access robustness of $\mathcal{H}$, we have

$$\begin{aligned}
&\mathsf{negl}(\lambda)\\
\geq&\Pr\Big[\mathcal{H}.\mathsf{Eval}(h, h(X_1'), h(X_2')) \neq \mathsf{SSD}^t(X_1', X_2')\Big]\\
\geq&\Pr\Big[\mathcal{H}.\mathsf{Eval}(h, h(X_1'), h(X_2')) \neq \mathsf{SSD}^t(X_1', X_2') \;\Big|\; |X_1' \triangle X_2'| = |X_1 \triangle X_2|\Big]\\
&\cdot \Pr\Big[|X_1' \triangle X_2'| = |X_1 \triangle X_2|\Big]\\
=&\Pr\Big[\mathcal{H}.\mathsf{Eval}(h, h(X_1'), h(X_2')) \neq \mathsf{SSD}^t(X_1, X_2) \;\Big|\; |X_1' \triangle X_2'| = |X_1 \triangle X_2|\Big]\\
&\cdot \Pr\Big[|X_1' \triangle X_2'| = |X_1 \triangle X_2|\Big]
\end{aligned}$$

where the last equality follows from the fact that $|X_1' \triangle X_2'| = |X_1 \triangle X_2|$ implies that $\mathsf{SSD}^t(X_1', X_2') = \mathsf{SSD}^t(X_1, X_2)$. Thus the claim follows. □

Claim 9. *If $\mathcal{F}$ is collision resistant, it holds that*

$$\Pr\big[|X_1' \triangle X_2'| \neq |X_1 \triangle X_2|\big] \leq \mathsf{negl}(\lambda)\,.$$

Proof (Claim 9). Note that, since f is a function, it must hold that $|X_1' \cup X_2'| \leq |X_1 \cup X_2|$. Further, if $|X_1' \cup X_2'| = |X_1 \cup X_2|$ then it must hold that $|X_1' \cap X_2'| \geq |X_1 \cap X_2|$. By definition of symmetric set difference it therefore holds that

$$\begin{aligned}
&\Pr\big[|X_1' \triangle X_2'| \neq |X_1 \triangle X_2|\big]\\
\leq&\Pr\big[|X_1' \cup X_2'| < |X_1 \cup X_2| \vee |X_1' \cap X_2'| > |X_1 \cap X_2|\big]\\
=&\Pr\big[\exists\, x_1, x_2 \in X_1 \cup X_2 \colon x_1 \neq x_2 \wedge f(x_1) = f(x_2)\big].
\end{aligned}$$

Since $\mathcal{F}$ is a family of collision resistant hash functions, the probability that $\mathcal{A}$ finds a collision is negligible and thus the claim follows. □

Combining Equation 3 with Claims 8 and 9, it thus follows that

$$\Pr[\mathsf{Eval}(h', h'(X_1), h'(X_2)) \neq \mathsf{SSD}^t(X_1, X_2)] \leq \mathsf{negl}(\lambda)$$

and the theorem follows. □

We obtain the following Corollary by combining Theorems 7 and 4.

Corollary 10. *Let* GGen *be a bilinear group generation algorithm that generates bilinear groups of prime order $q = q(\lambda)$ with $q > 2^\lambda$ relative to which the n-SBDL assumption holds. Let $\mathcal{F} = \{f : \{0,1\}^\ell \to \{0,1\}^\lambda\}$ be a collision resistant hash function family. Then for any $n = \mathsf{poly}(\lambda)$, and any $t < \frac{n(\ell - \log n)}{\log q(\lambda)} - 1$ there exists a $\frac{(t+1)\log q(\lambda)}{\log \binom{2^\ell}{n}} \leq \frac{(t+1)\log q(\lambda)}{n(\ell - \log n)}$-compressing direct-access robust PPH for for the two-input predicate SSD^t and domain $\mathcal{P}_n(\{0,1\}^\ell)$.*

4 PPH for Hamming Distance

In this section, we construct a PPH function for the hamming distance predicate. To hash a string $x \in \{0,1\}^n$, we apply our PPH function for the symmetric set difference predicate to the set encoding S_x of x.

Theorem 11. *Let $\mathcal{H}_\lambda = \{h : \mathcal{P}_n([2n]) \to Y\}$ be an η-compressing direct-access robust property preserving hash function family for SSD^{2t}. Then the following construction $\mathcal{H}'$ is a $\eta \cdot \frac{\log\binom{2n}{n}}{n} \leq \eta \cdot (1 + \log e)$-compressing direct-access robust PPH for HAM^t and domain $\{0,1\}^n$. $\mathcal{H}'$ is defined by $(\mathsf{Sample}', \mathsf{Hash}', \mathsf{Eval}')$ with $\mathsf{Sample}' = \mathsf{Sample}$, $\mathsf{Hash}'(x) := \mathsf{Hash}(S_x)$, and $\mathsf{Eval}' = \mathsf{Eval}$.*

Proof. Let $\mathcal{A}$ be an arbitrary PPT adversary against the direct-access robustness of $\mathcal{H}'$. We have that

$$\begin{aligned}
&\Pr[\mathsf{Eval}'(h', h'(x_1), h'(x_2)) \neq \mathsf{HAM}^t(x_1, x_2)] \\
=&\Pr[\mathsf{Eval}(h, h(S_{x_1}), h(S_{x_2})) \neq \mathsf{HAM}^t(x_1, x_2)] && (4)\\
=&\Pr[\mathsf{Eval}(h, h(S_{x_1}), h(S_{x_2})) \neq \mathsf{SSD}^{2t}(S_{x_1}, S_{x_2})] && (5)\\
\leq&\mathsf{negl}(\lambda) && (6)
\end{aligned}$$

where Equation 4 follows from the definition of $\mathcal{H}'$, Equation 5 follows from the fact that by Lemma 1 for any $x, y \in \{0,1\}^n$ it holds that $d(x_1, x_2) > t \iff |S_{x_1} \triangle S_{x_2}| > 2t$. Finally Equation 6 follows from the direct-access robustness of the underlying property preserving hash function family $\mathcal{H}$.

The inputs to the hash functions are of length n and are first mapped to elements of $\mathcal{P}_n([2n])$ before being hashed with an η-compressing function. The total compression is thus

$$\eta \cdot \frac{\log\binom{2n}{n}}{n} \leq \eta \cdot \frac{\log\left(\frac{e \cdot 2n}{n}\right)^n}{n} = \eta \cdot \log 2e = \eta \cdot (1 + \log e)$$

as claimed. □

Combining Theorems 11 and 4, we immediately get the following Corollary.

Corollary 12. *Let GGen be a bilinear group generation algorithm that generates bilinear groups of prime order $q = q(\lambda)$ with $q > 2^\lambda$ relative to which the n-SBDL assumption holds. Then for any $n = \mathsf{poly}(\lambda)$, and any $t < \frac{n}{2\log q(\lambda)} - \frac{1}{2}$ there exists a $\frac{(2t+1)\log q(\lambda)}{n}$-compressing direct-access robust PPH for for the two-input predicate HAM^t and domain $\{0,1\}^n$.*

4.1 Generalization to Different Alphabets

Previously, we have defined Hamming distance specifically for binary strings. This notion, however, as well as the corresponding predicate, can easily be generalized to strings over an arbitrary alphabet Σ. Let Σ be an alphabet and let $x, y \in \Sigma^n$ be strings. The Hamming distance between the two strings is the number of indices $i \in [n]$, such that $x_i \neq y_i$, formally $d(x, y) = |\{i \in [n] \mid x_i \neq y_i\}|$.

Using this generalized definition of Hamming distance, it is straightforward to generalize the Hamming predicate defined in Definition 2 to a predicate $\mathsf{HAM}^{\Sigma,t}$ for strings over an arbitrary alphabet Σ.

To generalize the construction from Theorem 11 to this predicate, we merely need to define a set-encoding for strings over Σ. Let $\Sigma = \{a_1, \ldots, a_\ell\}$ be an alphabet of size ℓ and let $x = x_i \ldots x_n = a_{i_1} \ldots a_{i_n} \in \Sigma^n$ be an arbitrary string over Σ. We define the set encoding of x as $S_x = \{\ell \cdot j - i_j \mid j \in [n]\}$. Using this set encoding in the construction from Theorem 11 immediately gives us a PPH function for $\mathsf{HAM}^{\Sigma,t}$ as stated in the following.

Proposition 13. *Let $\mathcal{H} = \{h : \mathcal{P}_n([\ell n]) \to Y\}$ be an η-compressing direct-access robust property preserving hash function family for SSD^{2t}. Then the following construction $\mathcal{H}'$ is a $\eta \cdot \frac{\log\binom{\ell n}{n}}{\log \ell^n} \leq \eta \cdot (1 + \frac{\log e}{\log \ell})$-compressing direct-access robust PPH for $\mathsf{HAM}^{\Sigma,t}$ and domain Σ^n. $\mathcal{H}'$ is defined by $(\mathsf{Sample}', \mathsf{Hash}', \mathsf{Eval}')$ with $\mathsf{Sample}' = \mathsf{Sample}$, $\mathsf{Hash}'(x) := \mathsf{Hash}(S_x)$, and $\mathsf{Eval}' = \mathsf{Eval}$.*

The proof easily follows from the proof of Theorem 11, by extending Lemma 1 to strings over arbitrary alphabets. Note, that the proof of Lemma 1 already proves this stronger statement.

5 PPH for Multi-Input Predicates

In this section, we show how to extend our constructions to the multi-input intersection predicate, which we introduce below. The basic idea underlying our construction is reminiscent to an idea used by Ghosh and Simkin [GS19a][6] for constructing interactive protocols that are secure against semi-honest adversaries for the so-called multiparty threshold private set intersection problem. Since we consider an active adversary and would like to construct a non-interactive primitive, our setting is quite a bit more challenging and requires a more intricate security analysis.

Definition 8 (Intersection Predicate). *For sets $X_1, \ldots, X_\ell \in \mathcal{P}_n(U)$ of size n with elements from the universe U and threshold $t > 0$, the multi-input set intersection predicate is defined as*

$$\mathsf{INT}^t_\ell(X_1, \ldots, X_\ell) = \begin{cases} 1 & \text{if } |X_1 \cap \cdots \cap X_\ell| > n - t \\ 0 & \text{Otherwise} \end{cases}$$

Before presenting our construction in this section, we observe that the symmetric set difference and the intersection predicate are equivalent for the two-input setting.

Proposition 14. *For all $n \in \mathbb{N}$, for all sets $X, Y \in \mathcal{P}_n(U)$ of size n with elements from the universe U and for all $t \in \mathbb{N}$, it holds that $\mathsf{INT}^t_2(X, Y) = 1 - \mathsf{SSD}^{2t}(X, Y)$.*

Proof. Let X and Y be two sets of size n with elements from an arbitrary universe U. We observe that

$$\begin{aligned} |X \triangle Y| =& |(X \setminus Y) \cup (Y \setminus X)| \\ =& |(X \setminus (X \cap Y)) \cup (Y \setminus (X \cap Y))| \\ =& n - |X \cap Y| + n - |X \cap Y| \\ =& 2n - 2|X \cap Y| \end{aligned}$$

and therefore

$$\begin{aligned} \mathsf{SSD}^{2t}(X, Y) = 1 &\iff |X \triangle Y| \geq 2t \\ &\iff 2n - 2|X \cap Y| \geq 2t \\ &\iff n - t \geq |X \cap Y| \iff \mathsf{INT}^t_2(X, Y) = 0 \end{aligned}$$

and equivalently $\mathsf{SSD}^{2t}(X, Y) = 0 \iff \mathsf{INT}^t_2(X, Y) = 1$. □

5.1 PPH for the Intersection Predicate INT^t_ℓ

The intuition for our PPH function for INT^t_ℓ is as follows. Let $X_1, \ldots, X_\ell$ be sets encoded into polynomials $p_{X_1}(z), \ldots, p_{X_\ell}(z)$ over a field $\mathbb{Z}_q$ of prime order. Let $W = X_1 \cap \cdots \cap X_\ell$ be the intersection of those sets and let $c_1, \ldots, c_\ell$ be field elements, then

$$\begin{aligned} & \frac{c_1 \cdot p_{X_1}(z) + \cdots + c_{\ell-1} \cdot p_{X_{\ell-1}}(z)}{c_1 \cdot p_{X_\ell}(z)} \\ = & \frac{p_W(z) \left(c_1 \cdot p_{X_1 \setminus W}(z) + \cdots + c_{\ell-1} \cdot p_{X_{\ell-1} \setminus W}(z)\right)}{p_W(z) \cdot c_\ell \cdot p_{X_\ell \setminus W}(z)} \\ = & \frac{c_1 \cdot p_{X_1 \setminus W}(z) + \cdots + c_{\ell-1} \cdot p_{X_{\ell-1} \setminus W}(z)}{c_\ell \cdot p_{X_\ell \setminus W}(z)} \end{aligned}$$

[6] Their multiparty protocols can be found in the extended abstract [GS19b] on ePrint.

If $|W| > n - t$, then for each $i \in [\ell]$ the degree of $p_{X_i \setminus W}(z)$ is upper bounded by t. This implies that the degree of the two polynomials in the numerator and denominator are upper bounded by t respectively, resulting in an upper bound of $2t$ for the total degree of the rational function. This is stated formally in the following lemma.

Lemma 15. *Let $n, N \in \mathbb{N}$ such that $n < N$ and let $\mathbb{Z}_q$ be a field of prime order $q > N$. For all $X_1, \ldots, X_\ell \in \mathcal{P}_n([N])$ and all $c_1, \ldots, c_\ell \in \mathbb{Z}_q^*$ it holds that*

$$2\Big(n - \Big|\bigcap_{i\in[\ell]} X_i\Big|\Big) \geq \mathsf{tdeg}\left(\frac{\sum_{i\in[\ell-1]} c_i \cdot p_{X_i}}{c_\ell \cdot p_{X_\ell}}\right)$$

Proof. Let $W = X_1 \cap \cdots \cap X_\ell$ be the intersection of the sets. We have

$$p_{X_\ell}(z) = \prod_{x\in X_\ell}(z-x) = \Big(\prod_{x\in X_\ell \setminus W}(z-x)\Big) \cdot \Big(\prod_{x\in W}(z-x)\Big) = p_{X_\ell \setminus W}(z) \cdot p_W(z)$$

and thus the degree of the denominator is at most $n - |W|$. Similarly, for any $1 \leq i \leq \ell - 1$, we have

$$c_i \cdot p_{X_i}(z) = c_i \cdot p_{X_i \setminus W}(z) \cdot p_W(z)$$

and thus the degree of each individual polynomial in the numerator is at most $n - |W|$. Since the sum of polynomials of degrees $d_1, \ldots, d_{\ell-1}$ is a new polynomial of degree $\max(d_1, \ldots, d_\ell)$, the lemma follows. □

To obtain something equivalent to Lemma 2, i.e. that the degree of the rational function corresponds *exactly* to $n-t$, we would like to argue that if $|W| \leq n-t$, then the degree of numerator and denominator are also *at least* t. However, this is not necessarily the case. Even though, after factoring out $p_W(z)$ the remaining polynomials $p_{X_i \setminus W}(x)$ no longer share any common roots, the *sum of polynomials* in the numerator *could* share an additional root with the numerator. However, by choosing the c_i with a random oracle and choosing our parameters appropriately, we can ensure that no efficient algorithm will be able to *find* such a combination with non-negligible probability. We formally state the following lemma.

Lemma 16. *Let $n, N \in \mathbb{N}$ with $n < N$ and let $\delta \geq \lambda + \ell \log^2 \lambda + \log N + 1$. Let $\mathbb{Z}_q$ be a field of prime order $q > 2^\delta$ and let $R : \mathcal{P}_n([N]) \to \mathbb{Z}_q^*$ be a random oracle. Then for any PPT algorithm $\mathcal{A}$ it holds that*

$$\Pr\left[2\Big(n - \Big|\bigcap_{i\in[\ell]} X_i\Big|\Big) > \mathsf{tdeg}\left(\frac{\sum_{i\in[\ell-1]} R(X_i) \cdot p_{X_i}}{R(X_\ell) \cdot p_{X_\ell}}\right)\right] \leq \mathsf{negl}(\lambda)\,,$$

where the randomness is taken over $(X_1, \ldots, X_\ell) \leftarrow \mathcal{A}^{R(\cdot)}(1^\lambda)$ and the choice of the random oracle.

Proof. Denote by $(X_1, \ldots, X_\ell)$ the output of $\mathcal{A}$ and by $W = \{w \mid p_{X_1}(w) = \cdots = p_{X_{\ell-1}}(w) = 0\}$ the set of common roots of the polynomials *in the numerator*. We first note, that the degree of the rational function can only be smaller than $(n - |\bigcap_{i\in[\ell]} X_i|)$, if an additional root of p_{X_1} cancels out. For this to be the case, the sum in the numerator must have a root $z \in [N] \setminus W$. To prove the lemma, it thus suffices to show that

$$\Pr\Big[\sum_{i\in[\ell-1]} R(X_i) \cdot p_{X_i}(z) = 0 \wedge \exists i \in [\ell-1].\ p_{X_i}(z) \neq 0\Big] \leq \mathsf{negl}(\lambda)\,.$$

Denote by Q the set of queries made to the random oracle R before $\mathcal{A}$ produces its output. For any fixed $z \in [N] \setminus W$, and any index i it holds that

$$\Pr\Big[\sum_{j\in[\ell-1]} R(X_j) \cdot p_{X_j}(z) = 0 \;\Big|\; \exists\, X_i \notin Q.\ p_{X_i}(z) \neq 0\Big]$$

$$= \Pr\Big[R(X_i) = -p_{X_i}^{-1}(z) \cdot \sum_{j \in [\ell-1] \setminus \{i\}} R(X_j) \cdot p_{X_j}(z) \;\Big|\; \exists\, X_i \notin Q.\ p_{X_i}(z) \neq 0\Big] \leq 2^{-\delta},$$

since the left-hand side is an independently and uniformly distributed element of $\mathbb{Z}_q^*$. By a union bound this gives us the following probability that there exists any such $z \in [N] \setminus W$ and $X_i \notin Q$:

$$\begin{aligned}
&\Pr\Big[\exists\, z \in [N] \setminus W.\ \sum_{j \in [\ell-1]} R(X_j) \cdot p_{X_j}(z) = 0 \wedge \exists\, X_i \notin Q.\ p_{X_i}(z) \neq 0\Big] \\
&\leq \sum_{z \in [N] \setminus W} \Pr\Big[\sum_{j \in [\ell-1]} R(X_j) \cdot p_{X_j}(z) = 0 \wedge \exists\, X_i \notin Q.\ p_{X_i}(z) \neq 0\Big] \\
&\leq \sum_{z \in [N] \setminus W} \Pr\Big[\sum_{j \in [\ell-1]} R(X_j) \cdot p_{X_j}(z) = 0 \;\Big|\; \exists\, X_i \notin Q.\ p_{X_i}(z) \neq 0\Big] \\
&= \sum_{z \in [N] \setminus W} 2^{-\delta} \leq N \cdot 2^{-\delta}.
\end{aligned}$$

Thus we can conclude that for any z and any $(X_1, \ldots, X_\ell)$ the adversary has to query R on all polynomials that are not vanishing at z to have any hope of succeeding at that evaluation point. At this point, the adversary's task is effectively reduced to finding a sequence $(X_1, \ldots, X_k) \in Q^k$ of length $k \in [\ell - 1]$ such that $p_X(z) \neq 0$ for all X_i, but

$$\sum_{i \in [k]} R(X_i) \cdot p_{X_i}(z) = 0.$$

Given such a sequence, the adversary can then win by simply "filling up" the sequence to length $\ell - 1$ using sets corresponding to polynomials that vanish at z.

Since $\mathcal{A}$ runs in polynomial time, there exists a $\mu = \mathsf{poly}(\lambda)$ such that $|Q| = \mu$. Let Y_i denote the ith query made by $\mathcal{A}$ and let $Q_i = \{Y_1, \ldots, Y_i\} \subseteq Q$ denote the set of the first i queries.

Fix an arbitrary $z \in [N]$ and consider the set $Z_i = \{R(Y) \cdot p_Y(z) \mid Y \in Q_i \wedge p_Y(z) \neq 0\}$ with $|Z_i| \leq |Q_i| = i$, which is a set of independent uniformly random elements of $\mathbb{Z}_q^*$ because R is a random oracle and none of the involved polynomials is 0 at point z. The number of sequences[7] of elements from Z_i of length at most $\ell - 1$ can be bounded as

$$\Big|\bigcup_{k \in [\ell-1]} Z_i^k\Big| \leq \sum_{k \in [\ell-1]} i^k \leq 2i^{\ell-1},$$

Assume that a sequence that sums up to zero does *not* exist in Z_{i-1}. Then for each of those sequences of Z_i elements, there is *at most* one value of $R(Y_i) \cdot p_{Y_i}(z)$ that would make the sequence sum up to zero. Let ZERO be the event that at least one sequence summing up to zero occurs in Z_μ and let ZERO_i be the event that the first such sequence occurs after the ith query. Then by the above observation, we have

$$\begin{aligned}
\Pr[\mathsf{ZERO}] = \sum_{i=1}^{\mu} \Pr[\mathsf{ZERO}_i] &\leq \sum_{i=1}^{\mu} \Pr\Big[\bigwedge_{j \in [i-1]} \neg \mathsf{ZERO}_j\Big] \cdot \frac{2i^{\ell-1}}{2^\delta} \\
&\leq 2^{1-\delta} \cdot \sum_{i=1}^{\mu} i^{\ell-1} \\
&\leq 2^{1-\delta} \mu^\ell \\
&\leq 2^{1-\delta} \lambda^{\ell \log \lambda} = 2^{1+\ell \log^2 \lambda - \delta}
\end{aligned}$$

[7] Taking into account the commutativity of addition in $\mathbb{F}$, many of these sequences are actually equivalent. It would be sufficient to count the number of possible *multi-sets* instead. However, counting sequences is an upper bound on this actual number and gives a simpler, though slightly worse, bound for δ.

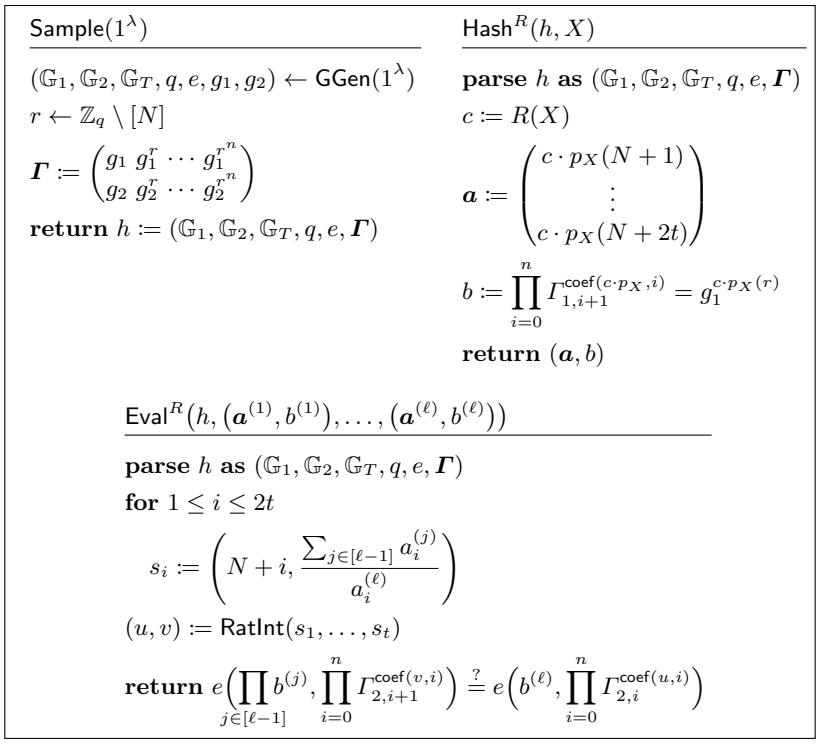

Fig. 3. A family of direct-access robust PPHs for INT_ℓ^t.

where the last inequality follows from the fact that $\mu \le \lambda^{\log \lambda}$ for large enough λ. By a union bound over $z \in [N]$, we get that

$$\Pr\Big[\exists z \in [N].\ \exists\ (X_1, \ldots, X_k) \in \bigcup_{j \in [\ell-1]} Q^j.\ \sum_{i \in [k]} p_{X_i}(z) = 0\Big] = 2^{\log N + 1 + \ell \log^2 \lambda - \delta}.$$

Note that this event is exactly the event that the adversary can find the desired sequence described above. Since by the lemma statement, $\delta \ge \lambda + (\ell+1)\log^2 \lambda + \log N + 1$ the lemma follows. □

Equipped with these observations, our construction will now be a natural extension of our previous constructions for the two-input case. The proof of Theorem 17 will therefore mirror the proof of Theorem 4 closely.

Theorem 17. *Let $n = \mathsf{poly}(\lambda)$, $N \in \mathbb{N}$ with $n \le N \le 2^\lambda$. Let GGen be a bilinear group generation algorithm that generates bilinear groups of prime order $q(\lambda) > 2^\delta$, where $\delta \ge \lambda + \ell \log^2 \lambda + \log N + 1$ and let $R : \mathcal{P}_n([N]) \to \mathbb{Z}_q^*$ be a random oracle. Then for any any $t < \frac{n(\log N - \log n)}{2 \log q(\lambda)} - \frac{1}{2}$ the construction in Figure 3 is a $\frac{(2t+1)\log q(\lambda)}{n(\log N - \log n)}$-compressing direct-access robust PPH for for the multi-input predicate INT_ℓ^t and domain $\mathcal{P}_n([N])$ if the n-SBDL assumption holds relative to GGen.*

Proof. Let $\mathcal{A}$ be an arbitrary PPT adversary against the direct access robustness of $\mathcal{H}$. We have

$$\begin{aligned} & \Pr[\mathsf{Eval}(h, h(X_1), \ldots, h(X_\ell)) \neq \mathsf{INT}_\ell^t(X_1, \ldots, X_\ell)] \\ = \quad & \Pr[\mathsf{Eval}(h, h(X_1), \ldots, h(X_\ell)) = 0 \mid \mathsf{INT}_\ell^t(X_1, \ldots, X_\ell) = 1] \\ & \cdot \Pr[\mathsf{INT}_\ell^t(X_1, \ldots, X_\ell) = 1] \end{aligned}$$

$$+ \Pr[\mathsf{Eval}(h, h(X_1), \ldots, h(X_\ell)) = 1 \mid \mathsf{INT}^t_\ell(X_1, \ldots, X_\ell) = 0] \cdot \Pr[\mathsf{INT}^t_\ell(X_1, \ldots, X_\ell) = 0],$$

where the probabilities are taken over $h \leftarrow \mathsf{Sample}(1^\lambda)$ and $(X_1, \ldots, X_\ell) \leftarrow \mathcal{A}(h)$. We consider the two cases separately.

Claim 18. $\Pr[\mathsf{Eval}(h, h(X_1), \ldots, h(X_\ell)) = 0 \mid \mathsf{INT}^t_\ell(X_1, \ldots, X_\ell) = 1] = 0$

Proof (Claim 18). Let $c_i = R(X_i)$. Since $\mathsf{INT}^t_\ell(X_1, \ldots, X_\ell) = 1$, it holds that $|X_1 \cap \cdots \cap X_\ell| > n - t$ and by Lemma 15 that

$$\mathsf{tdeg}\left(\frac{\sum_{i\in[\ell-1]} c_i \cdot p_{X_i}}{c_\ell \cdot p_{X_\ell}}\right) < 2t.$$

By Proposition 3, it follows that the rational function can thus be uniquely (up to equivalences) interpolated from $2t$ points. We observe that for $1 \le i \le 2t$ it holds that $c_\ell \cdot p_{X_\ell}(N+i) \neq 0$, since the roots of p_{X_ℓ} are in the interval $[N]$ by construction and $c_1 \in \mathbb{Z}_q^*$. Therefore,

$$s_i = \frac{\sum_{j\in[\ell-1]} c_j \cdot p_{X_j}(N+i)}{c_\ell \cdot p_{X_\ell}(N+i)}$$

are well-defined and thus

$$\frac{\sum_{j\in[j-1]} c_j \cdot p_{X_j}}{c_\ell \cdot p_{X_\ell}} = \frac{u}{v} \tag{7}$$

where u/v is the rational function computed by RatInt in $\mathsf{Eval}(h, h(X_1), \ldots, h(X_\ell))$. Finally we observe that

$$\begin{aligned}
& e\Big(\prod_{j\in[\ell-1]} b^{(j)}, \prod_{i=0}^{n} \Gamma_{2,i+1}^{\mathsf{coef}(v,i)}\Big) = e\Big(b^{(\ell)}, \prod_{i=0}^{n} \Gamma_{2,i}^{\mathsf{coef}(u,i)}\Big) \\
\iff & e\Big(g_1^{\sum_{j\in[\ell-1]} c_j \cdot p_{X_j}(r)}, g_2^{v(r)}\Big) = e\Big(g_1^{c_\ell \cdot p_{X_\ell}(r)}, g_2^{u(r)}\Big) \\
\iff & e(g_1, g_2)^{\left(\sum_{j\in[\ell-1]} c_j \cdot p_{X_j}(r)\right)\cdot v(r)} = e(g_1, g_2)^{c_\ell \cdot p_{X_\ell}(r)\cdot u(r)} \\
\iff & \Big(\sum_{j\in[\ell-1]} c_j \cdot p_{X_j}(r)\Big) \cdot v(r) = c_\ell \cdot p_{X_\ell}(r) \cdot u(r) \\
\iff & \frac{\sum_{j\in[\ell-1]} c_j \cdot p_{X_j}(r)}{c_\ell \cdot p_{X_\ell}(r)} = \frac{u(r)}{v(r)},
\end{aligned}$$

which due to Equation 7 is always true and thus Eval always returns 1 in this case. □

Claim 19. *If the n-SBDL assumption holds relative to* GGen*, then*

$$\begin{aligned}
& \Pr[\mathsf{Eval}(h, h(X_1), \ldots, h(X_\ell)) = 1 \mid \mathsf{INT}^t_\ell(X_1, \ldots, X_\ell) = 0] \cdot \Pr[\mathsf{INT}^t_\ell(X_1, \ldots, X_\ell) = 0] \\
& \le \mathsf{negl}(\lambda).
\end{aligned}$$

Proof (Claim 19). Since $\mathsf{INT}^t_\ell(X_1, \ldots, X_\ell) = 0$, it must hold that $0 \le |X_1 \cap \cdots \cap X_\ell| \le n - t$. By Lemma 16, since $\mathcal{A}$ is a PPT algorithm, this means that except with negligible probability

$$2t \le \mathsf{tdeg}\left(\frac{\sum_{i\in[\ell-1]} c_i \cdot p_{X_i}}{c_\ell \cdot p_{X_\ell}}\right) \le 2n.$$

On the other hand, by construction u/v is the rational function of total degree at most $2t-1$ uniquely determined by $s_1, \ldots, s_{2t}$. It must therefore hold that

$$\frac{u}{v} \neq \frac{\sum_{i\in[\ell-1]} c_i \cdot p_{X_\ell}}{c_\ell \cdot p_{X_\ell}}.$$

For the last inequality in $\mathsf{Eval}(h, h(X_1), \ldots, h(X_\ell))$ to hold, $(\sum_{i\in[\ell-1]} c_i \cdot p_{X_\ell})/(c_1 \cdot p_{X_\ell})$ and u/v must therefore be two *different* rational functions that agree on point r. This means that r must be one of the at most $n + (t-1)/2$ roots of the rational function

$$\frac{\sum_{i\in[\ell-1]} c_i \cdot p_{X_\ell}}{c_\ell \cdot p_{X_\ell}} - \frac{u}{v} = \frac{v \cdot \sum_{i\in[\ell-1]} c_i \cdot p_{X_\ell} - c_\ell \cdot p_{X_\ell} \cdot u}{c_\ell \cdot p_{X_\ell} \cdot v}.$$

Whenever $\mathcal{A}$ would be successful, we could therefore find r by testing the roots of the polynomial $v \cdot \sum_{i\in[\ell-1]} c_i \cdot p_{X_\ell} - c_\ell \cdot p_{X_\ell} \cdot u$. We give a formal reduction as follows:

$\mathcal{R}$ takes as input

$$\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, e, \boldsymbol{\Gamma} \coloneqq \begin{pmatrix} g_1 & g_1^r & \cdots & g_1^{r^n} \\ g_2 & g_2^r & \cdots & g_2^{r^n} \end{pmatrix}$$

and invokes $\mathcal{A}$ on $h \coloneqq (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, e, \boldsymbol{\Gamma})$ and receives $X_1, \ldots, X_\ell$.

The reduction then checks whether $\mathsf{INT}^t_\ell(X_1, \ldots, X_\ell) = 0$ and aborts otherwise. We denote this event as INT_0. Next $\mathcal{R}$ checks whether

$$\mathsf{tdeg}\left(\frac{\sum_{i\in[\ell-1]} c_i \cdot p_{X_i}}{c_\ell \cdot p_{X_\ell}}\right) \geq 2t$$

and again aborts otherwise. We denote this event as $\mathsf{TDEG}_{\geq 2t}$. Note, that as argued above, by Lemma 16

$$\Pr[\mathsf{TDEG}_{\geq} \mid \mathsf{INT}_0] \geq 1 - \mathsf{negl}(\lambda)\,.$$

If it has not aborted, $\mathcal{R}$ then computes (u, v) as in Eval and determines the set X of roots of the polynomial $v \cdot \sum_{i\in[\ell-1]} c_i \cdot p_{X_\ell} - c_\ell \cdot p_{X_\ell} \cdot u$. For each $r' \in X$, $\mathcal{R}$ checks whether $g_1^{r'} \stackrel{?}{=} \Gamma_{1,2}$ and returns r' if it holds. If the equality holds for no $r' \in X$, $\mathcal{R}$ aborts.

The reduction $\mathcal{R}$ essentially performs three steps, executing $\mathcal{A}$, checking the total degree of a rational function, and finding the roots of a polynomial. Each of those steps can be performed in polynomial time. It follows that $\mathcal{R}$ is PPT and must, by assumption, have a negligible success probability against the n-SBDL problem.

Note that r from the input of the reduction is distributed uniformly in $\mathbb{Z}_q$, while $\mathcal{A}$ expects r to be uniformly distributed in $\mathbb{Z}_q \setminus [N]$. However, since and $q > 2^\delta \geq 2^{\lambda+\ell\log^2\lambda+\log N+1} \geq N \cdot 2^{\lambda+\ell\log^2\lambda+1}$, it holds that $r \in \mathbb{Z}_q \setminus [N]$ with probability at least $1 - 2^{-\lambda-\ell\log^2\lambda-1}$. Furthermore, once we condition on $r \notin [N]$, the distribution of h is identical to the one expected by $\mathcal{A}$.

Now, observe that the reduction $\mathcal{R}$ is successful, if $\mathcal{A}$ outputs $X_1, \ldots, X_\ell$, such that INT_0 and $\mathsf{TDEG}_{\geq 2t}$ both occur and r is one of the roots of $v \cdot \sum_{i\in[\ell-1]} c_i \cdot p_{X_\ell} - c_\ell \cdot p_{X_\ell} \cdot u$. As argued above, conditioned on the first two, the latter must be true, if $\mathsf{Eval}(h, h(X_1), \ldots, h(X_\ell)) = 1$. Therefore, it holds that

$$\begin{aligned}
&\mathsf{negl}(\lambda) \\
\geq &\Pr\left[r = \mathcal{R}\left(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, e, \begin{pmatrix} g_1 & g_1^r & \cdots & g_1^{r^n} \\ g_2 & g_2^r & \cdots & g_2^{r^n} \end{pmatrix}\right)\right] \\
\geq &\Pr\big[r \notin [N]\big] \cdot \Pr\left[r = \mathcal{R}\left(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, e, \begin{pmatrix} g_1 & g_1^r & \cdots & g_1^{r^n} \\ g_2 & g_2^r & \cdots & g_2^{r^n} \end{pmatrix}\right) \;\middle|\; r \notin [N]\right] \\
\geq &(1 - \mathsf{negl}(\lambda)) \cdot \Pr[\mathsf{Eval}(h, h(X_1), \ldots, h(X_\ell)) = 1 \mid \mathsf{INT}_0, \mathsf{TDEG}_{\geq 2t}] \\
&\cdot \Pr[\mathsf{TDEG}_{\geq 2t} \mid \mathsf{INT}_0] \cdot \Pr[\mathsf{INT}_0]
\end{aligned}$$

$$
\begin{aligned}
&\geq \Pr[\mathsf{Eval}(h, h(X_1), \ldots, h(X_\ell)) = 1 \mid \mathsf{INT}_0, \mathsf{TDEG}_{\geq 2t}] \cdot \Pr[\mathsf{TDEG}_{\geq 2t} \mid \mathsf{INT}_0] \cdot \Pr[\mathsf{INT}_0] \\
&\quad - \mathsf{negl}(\lambda) \\
&\geq \Pr[\mathsf{Eval}(h, h(X_1), \ldots, h(X_\ell)) = 1 \mid \mathsf{INT}_0] \cdot \Pr[\mathsf{INT}_0] - \Pr[\neg \mathsf{TDEG}_{\geq 2t} \mid \mathsf{INT}_0] - \mathsf{negl}(\lambda) \\
&\geq \Pr[\mathsf{Eval}(h, h(X_1), \ldots, h(X_\ell)) = 1 \mid \mathsf{INT}_0] \cdot \Pr[\mathsf{INT}_0] - \mathsf{negl}(\lambda)
\end{aligned}
$$

and the claim follows. □

Using Claims 18 and 19 we can thus conclude that

$$
\Pr[\mathsf{Eval}(h, h(X_1), \ldots, h(X_\ell)) \neq \mathsf{INT}^t_\ell(X_1, \ldots, X_\ell)] \leq 0 + \mathsf{negl}(\lambda) = \mathsf{negl}(\lambda)\,.
$$

Therefore, $\mathcal{H}$ is direct access robust as claimed. It remains to show that it is also compressing. The domain of the hash function is $\mathcal{P}_n([N])$, the codomain is $\mathbb{Z}^t_{q(\lambda)} \times \mathbb{G}_1$. It follows that the compression factor is

$$
\eta \leq \frac{\log |\mathbb{Z}^{2t}_{q(\lambda)} \times \mathbb{G}_1|}{\log |\mathcal{P}_n([N])|} = \frac{\log q(\lambda)^{2t+1}}{\log \binom{N}{n}} \leq \frac{\log q(\lambda)^{2t+1}}{\log (\frac{N}{n})^n} = \frac{(2t+1)\log q(\lambda)}{n(\log N - \log n)}
$$

as claimed. The construction is thus compressing, if

$$
\frac{(2t+1)\log q(\lambda)}{n(\log N - \log n)} < 1 \iff t < \frac{n(\log N - \log n)}{2 \log q} - \frac{1}{2}.
$$

□

6 Lower Bounds

In this section, we show that the compression rate of our constructions for the SSD^t and INT^t_2 predicates are close to optimal. In a similar fashion to BLV, we prove our lower bound on the size of a hash value by drawing connections to one-round communication complexity lower bounds. More precisely, we prove our lower bound by making use of the following lower bound for the set disjointness problem.

Theorem 20 ([DKS12]). *For a universe U, let $X, Y \subseteq \mathcal{P}_n(U)$. Let the set disjointness predicate be defined as follows:*

$$
\mathsf{DISJ}(X, Y) = \begin{cases} 1 & \text{if } X \cap Y = \emptyset \\ 0 & \text{Otherwise} \end{cases}
$$

For $n < \sqrt{|U|}$, the one-way randomized communication complexity of $\mathsf{DISJ}(X, Y)$ in the common random string model is $\Omega(n \log n)$.

In contrast to BLV, who prove the non-existence of PPH functions for certain parameters, we prove a lower bound on the size of the hash value for parameters where PPH functions are feasible.

Theorem 21. *Let $\mathcal{H} = \{h : \mathcal{P}_n(U) \to Y\}$ be a family of direct-access robust PPH functions for the symmetric set difference predicate SSD^t for some universe U with $|U| > t^2/4 + n - t/2$. Then,*

$$
\log |Y| \in \Omega(t \log t).
$$

Proof. We assume without loss of generality, that t is even.[8] Fix an arbitrary set $S \in \mathcal{P}_{n-t/2}(U)$. We prove the stated theorem by using $\mathcal{H}$ to construct a communication efficient one-round protocol for the set disjointness problem for input sets of size $t/2$ from the universe $U' = U \setminus S$. Let R be the common random string that the

[8] Note that for sets of equal size, the symmetric set difference is always even and therefore $\mathsf{SSD}^{2i-1} = \mathsf{SSD}^{2i}$ for all $i \in \mathbb{N}_+$.

parties can access in the set disjointness problem. Let $A, B \in \mathcal{P}_{n'}(U')$ be the input sets of the two parties. The protocol proceeds as follows:

The parties define $A' = A \cup S$ and $B' = B \cup S$. We note that $|A'| = |B'| = n$ and that $\mathsf{SSD}^t(A', B') = 1$ if and only if $A \cap B = \emptyset$. I.e., $\mathsf{SSD}^t(A', B') = \mathsf{DISJ}(A, B)$. Both parties then sample a hash function $h \in \mathcal{H}$ using randomness R and security parameter n. We let party P_A holding A send $z = h(A')$ to party P_B holding B. Party P_B computes $b = \mathsf{Eval}(h, z, h(B'))$ and outputs b. Note, that A', B' are fixed before and independently of h. It follows from the direct access robustness of $\mathcal{H}$ that for any such a priori fixed A', B' it holds that

$$\Pr[\mathsf{Eval}(h, h(A'), h(B') = \mathsf{SSD}^t(A', B')] \geq 1 - \mathsf{negl}(n)$$

where the probability is taken over the random choice of $h \in \mathcal{H}$. It therefore holds that $\Pr[b = \mathsf{DISJ}^t(A, B)] \geq 1 - \mathsf{negl}(n)$.

Observe that by definition of U' and S, it holds that $|U'| = |U| - |S| = |U| - (n - t/2)$. By the condition on $|U|$ from the theorem statement, it thus follows that

$$|U'| > \frac{t^2}{4} + n - t/2 - (n - t/2) = \frac{t^2}{4},$$

and thereby $\sqrt{|U'|} > t/2$. Since the protocol described above works for sets of size $t/2$, Theorem 20 therefore applies. The total communication of our protocol consists of $z \in Y$, thus by Theorem 20 we have that $\log |Y| \in \Omega(t \log t)$. □

Via the equivalence of the SSD^t and INT^t predicate proven in Proposition 14, we immediately also get the following lower bound on size of a hash value of a PPH function for INT_2^t.

Corollary 22. *Let $\mathcal{H} = \{h : \mathcal{P}_n(U) \to Y\}$ be a family of direct-access robust PPH functions for the two-input intersection predicate INT_2^t for some universe U with $|U| > t^2 + n - t$. Then,*

$$\log |Y| \in \Omega(t \log t).$$

References

BB04. Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity based encryption without random oracles. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 223–238, Interlaken, Switzerland, May 2–6, 2004. Springer, Heidelberg, Germany.

BEJWY20. Omri Ben-Eliezer, Rajesh Jayaram, David P Woodruff, and Eylon Yogev. A framework for adversarially robust streaming algorithms. In *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 63–80, 2020.

Blo70. Burton H Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.

BLV19. Elette Boyle, Rio LaVigne, and Vinod Vaikuntanathan. Adversarially robust property-preserving hash functions. In Avrim Blum, editor, *ITCS 2019: 10th Innovations in Theoretical Computer Science Conference*, volume 124, pages 16:1–16:20, San Diego, CA, USA, January 10–12, 2019. LIPIcs.

CPS19. David Clayton, Christopher Patton, and Thomas Shrimpton. Probabilistic data structures in adversarial environments. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *ACM CCS 2019: 26th Conference on Computer and Communications Security*, pages 1317–1334. ACM Press, November 11–15, 2019.

DKS12. Anirban Dasgupta, Ravi Kumar, and D Sivakumar. Sparse and lopsided set disjointness via information theory. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 517–528. Springer, 2012.

GOR11. Vipul Goyal, Adam O'Neill, and Vanishree Rao. Correlated-input secure hash functions. In Yuval Ishai, editor, *TCC 2011: 8th Theory of Cryptography Conference*, volume 6597 of *Lecture Notes in Computer Science*, pages 182–200, Providence, RI, USA, March 28–30, 2011. Springer, Heidelberg, Germany.

GS19a. Satrajit Ghosh and Mark Simkin. The communication complexity of threshold private set intersection. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part II*, volume 11693 of *Lecture Notes in Computer Science*, pages 3–29, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Heidelberg, Germany.

GS19b. Satrajit Ghosh and Mark Simkin. The communication complexity of threshold private set intersection. Cryptology ePrint Archive, Report 2019/175, 2019. https://eprint.iacr.org/2019/175.

HW13. Moritz Hardt and David P. Woodruff. How robust are linear sketches to adaptive inputs? In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th Annual ACM Symposium on Theory of Computing*, pages 121–130, Palo Alto, CA, USA, June 1–4, 2013. ACM Press.

IM98. Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *30th Annual ACM Symposium on Theory of Computing*, pages 604–613, Dallas, TX, USA, May 23–26, 1998. ACM Press.

MNS08. Ilya Mironov, Moni Naor, and Gil Segev. Sketching in adversarial environments. In Richard E. Ladner and Cynthia Dwork, editors, *40th Annual ACM Symposium on Theory of Computing*, pages 651–660, Victoria, BC, Canada, May 17–20, 2008. ACM Press.

MTZ03. Yaron Minsky, Ari Trachtenberg, and Richard Zippel. Set reconciliation with nearly optimal communication complexity. *IEEE Transactions on Information Theory*, 49(9):2213–2218, 2003.

Mut03. S. Muthukrishnan. Data streams: algorithms and applications. In *14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 413–413, Baltimore, MD, USA, January 12–14, 2003. ACM-SIAM.

NY15. Moni Naor and Eylon Yogev. Bloom filters in adversarial environments. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part II*, volume 9216 of *Lecture Notes in Computer Science*, pages 565–584, Santa Barbara, CA, USA, August 16–20, 2015. Springer, Heidelberg, Germany.

PR04. Rasmus Pagh and Flemming Friche Rodler. Cuckoo hashing. *J. Algorithms*, 51(2):122–144, May 2004.