Universal hash families and the leftover hash lemma, and applications to cryptography and computing

D.R. Stinson
Department of Combinatorics and Optimization
University of Waterloo
Waterloo Ontario, N2L 3G1, Canada
dstinson@uwaterloo.ca

Abstract

This paper is an expository treatment of the leftover hash lemma and some of its applications in cryptography and complexity theory.

1 Introduction

The technique of universal hashing, introduced in 1979 by Carter and Wegman [6], has become an essential tool in many areas of computer science, including derandomization, pseudorandom number generation and privacy amplification, to mention three specific applications. It has been observed that universal hash families are very closely related to combinatorial structures such as orthogonal arrays ([11]) and error-correcting codes ([15]), and we will frequently make use of these connections. (For a survey, see Stinson [21].)

Several random number generators related to strongly universal hash families have been shown to have desirable quasirandomness properites; see for example, [17]. (Quasirandomness provides a measure of how closely a given probability distribution approximates the uniform distribution.) We will give a self-contained, elementary treatment of this theory. The bounds on quasirandomness that we provide are exact, rather than asymptotic.

We also provide a thorough discussion of the so-called leftover hash lemma, which was proven by Impagliazzo, Levin and Luby in [12] (the term "leftover hash lemma" was first coined in Impagliazzo and Zuckerman [13]). We provide a simple combinatorial proof of this result, which is

JCMCC **42** (2002), pp. 3-31

also known as the "smoothing entropy theorem". We survey several consequences of this lemma, including the construction of extractors (which are used in derandomization of probabilistic algorithms) and quasirandom number generators (which are used in cryptography), as well as techniques of privacy amplification (another cryptographic application). Finally, we discuss how codes and orthogonal arrays can be used to provide simple constructions of these objects for various parameter situations of interest.

This paper is mainly expository in nature, and we include proofs of almost all the results in it. The remainder of the paper is organized as follows. In Section 2, we give several definitions and basic properties of different flavours of universal hash families. In Section 3, we recall several useful constructions of hash families. Section 4 presents, in an informal way, three applications of universal hash families: quasirandom number generation, privacy amplification, and derandomization. Section 5 presents the basic theory and definitions relating to concepts such as distance between probability distributions, distinguishability of probability distributions, quasirandomness of probability distributions, collision probability, and different types of entropies. In Section 6, we state and prove a basic combinatorial lemma concerning an important quasirandom property of strongly universal hash families, and examine the applications of this lemma to certain tyes of random number generators. In Section 7, we present the leftover hash lemma, which concerns quasirandom properties of δ-universal hash families, similar to those considered in the previous section. One application of the leftover hash lemma is studied in Section 8, namely, the concept of extractors, which are used for partial derandomization of probabilistic algorithms in the class BPP. Another application, privacy amplification, is studied in Section 9. Finally, we make some concluding remarks in Section 10.

2 Universal hash families

We begin by recalling some definitions of hash families.

- A $(D; N, M)$ *hash family* is a set $\mathcal{F}$ of D functions such that $f : X \to Y$ for each $f \in \mathcal{F}$, where $|X| = N$ and $|Y| = M$.
- A $(D; N, M)$-hash family, $\mathcal{F}$, is *δ-universal* ([20]) provided that for any two distinct elements $x_1, x_2 \in X$, there exist at most δD functions $f \in \mathcal{F}$ such that $f(x_1) = f(x_2)$. The parameter δ is often referred to as the *collision probability* of the hash family. We will use the notation δ-U as an abbreviation for δ-universal.
- A $(D; N, M)$-hash family, $\mathcal{F}$, is *strongly universal* ([6]) provided that,

for any two distinct elements $x_1, x_2 \in X$, and for any two (not necessarily distinct) elements $y_1, y_2 \in Y$, it holds that

$$|\{f \in \mathcal{F} : f(x_i) = y_i, i = 1, 2\}| = \frac{D}{M^2}.$$

We will use the notation SU as an abbreviation for strongly universal.

We will often depict a $(D; N, M)$ hash family, say $\mathcal{F}$, in the form of a $D \times N$ array of M symbols, where the rows are indexed by the functions in $\mathcal{F}$, the columns are indexed by the elements in X, and the entry in row f and column x of the array is $f(x)$ (for every $f \in \mathcal{F}$ and every $x \in X$). Each row of the array corresponds to one of the functions in the family. We denote this array by $\mathsf{A}(\mathcal{F})$ and call it *array representation* of the hash family $\mathcal{F}$. If $\mathcal{F}$ is a δ-U $(D; N, M)$ hash family, then in any two columns of $\mathsf{A}(\mathcal{F})$, it follows that there exist at most δD rows of $\mathsf{A}(\mathcal{F})$ such that the entries in the two given columns are equal.

Let Y be an alphabet of q symbols. An (n, K, d, q) *code* is a set $\mathcal{C}$ of K vectors (called *codewords*) in Y^n such that the Hamming distance between any two distinct vectors in $\mathcal{C}$ is at least d. If the code is *linear* (i.e., if q is a prime power, $Y = \mathbb{F}_q$, and $\mathcal{C}$ is a subspace of $(\mathbb{F}_q)^n$), then we will say that the code is an $[n, k, d, q]$ code, where $k = \log_q K$ is the *dimension* of the code.

The following equivalence was first observed by Bierbrauer, Johansson, Kabatianskii and Smeets in [3].

Theorem 2.1 *If there exists an (n, K, d, q) code, then there exists a $(1 - \frac{d}{n})$-U $(n; K, q)$ hash family. Conversely, if there exists an δ-U $(D; N, M)$ hash family, then there exists an $(D, N, D(1-\delta), M)$ code.*

Theorem 2.1 is proved by letting the codewords in the stated code correspond to the columns of $\mathsf{A}(\mathcal{F})$, where $\mathcal{F}$ is the stated hash family.

Example 2.1 The following $\frac{1}{3}$-U $(3; 9, 3)$ hash family $\{f_i : i \in \mathbb{Z}_3\}$ is equivalent to a $(3, 9, 2, 3)$ code:

	$(0,0)$	$(0,1)$	$(0,2)$	$(1,0)$	$(1,1)$	$(1,2)$	$(2,0)$	$(2,1)$	$(2,2)$
f_0 :	0	0	0	1	1	1	2	2	2
f_1 :	0	1	2	1	2	0	2	0	1
f_2 :	0	2	1	1	0	2	2	1	0

∎

An *orthogonal array* $\mathrm{OA}_\lambda(N, M)$ is a λM^2 by N array of M symbols such that, in any two columns of the array, each ordered pair of symbols

occurs in exactly λ rows. If $\mathcal{F}$ is an SU $(D; N, M)$-hash family, then it is immediate that $\mathsf{A}(\mathcal{F})$ is an $\mathrm{OA}_\lambda(N, M)$, where $\lambda = D/M^2$. The converse also holds, so we have the following theorem, which was first observed in [19].

Theorem 2.2 *An* SU $(D; N, M)$*-hash family is equivalent to an* $OA_\lambda(N, M)$*, where* $\lambda = D/M^2$.

Example 2.2 The following SU $(9; 3, 3)$ hash family $\{f_{i,j} : i, j \in \mathbb{Z}_3\}$ is equivalent to an $\mathrm{OA}_1(3, 3)$:

	0	1	2
$f_{0,0}$:	0	1	1
$f_{0,1}$:	1	2	2
$f_{0,2}$:	2	0	0
$f_{1,0}$:	1	1	0
$f_{1,1}$:	2	2	1
$f_{1,2}$:	0	0	2
$f_{2,0}$:	1	0	1
$f_{2,1}$:	2	1	2
$f_{2,2}$:	0	2	0

3 Some Constructions of Hash Families

We give several constructions of hash families in this section. The main idea in the following construction was used by Rao in 1947 ([18]) using the language of orthogonal arrays.

Theorem 3.1 *Let ℓ be a positive integer and let q be a prime power. Let $X \subseteq (\mathbb{F}_q)^\ell$ be any collection of pairwise linearly independent vectors over $\mathbb{F}_q$. For every $\vec{r} \in (\mathbb{F}_q)^\ell$, define a function $f_{\vec{r}} : X \to \mathbb{F}_q$ by the rule*

$$f_{\vec{r}}(\vec{x}) = \vec{r} \cdot \vec{x}.$$

Finally, define

$$\mathcal{F}(q, \ell, X) = \{f_{\vec{r}} : \vec{r} \in (\mathbb{F}_q)^\ell\}.$$

Then $\mathcal{F}(q, \ell, X)$ is an SU $(q^\ell; |X|, q)$*-hash family.*

Proof. Clearly we have a $(q^\ell; |X|, q)$-hash family. We need to prove that it is SU. Let $\vec{x_1}, \vec{x_2} \in (\mathbb{F}_q)^\ell$ (where $\vec{x_1} \neq \vec{x_2}$) and let $y_1, y_2 \in \mathbb{F}_q$. We want to count the number of vectors $\vec{r} \in (\mathbb{F}_q)^\ell$ such that

$$\vec{r} \cdot \vec{x_1} = y_1$$

and

$$\vec{r} \cdot \vec{x_2} = y_2.$$

Now $\vec{x_1}$ and $\vec{x_2}$ are linearly independent vectors over $\mathbb{F}_q$. If we denote $\vec{r} = (r_1, \ldots, r_\ell)$, we have a linearly independent system, in $\mathbb{F}_q$, of two equations in the ℓ unknowns $r_1, \ldots, r_\ell$. There are therefore exactly $q^{\ell-2}$ solutions for the vector $\vec{r}$, and the hash family is SU. □

We present a couple of corollaries of Theorem 3.1. The hash family constructed in Corollary 3.2 is equivalent to the classical desarguesian affine plane of order q.

Corollary 3.2 *Let q be a prime power. For $a, b \in \mathbb{F}_q$, define $f_{a,b} : \mathbb{F}_q \to \mathbb{F}_q$ by the rule*

$$f_{a,b}(x) = ax + b.$$

Then $\{f_{a,b} : a, b \in \mathbb{F}_q\}$ is an SU $(q^2; q, q)$-hash family.

Proof. Take $\ell = 2$ and let $X = \mathbb{F}_q \times \{1\}$. Then apply Theorem 3.1. □

The hash family presented in Corollary 3.3 was proposed in [5].

Corollary 3.3 *Let ℓ be a positive integer and let q be a prime power. Let $X = \{0,1\}^\ell \backslash \{(0, \ldots, 0)\}$. For every $\vec{r} \in (\mathbb{F}_q)^\ell$, define $f_{\vec{r}} : X \to \mathbb{F}_q$ by the rule*

$$f_{\vec{r}}(\vec{x}) = \vec{r} \cdot \vec{x}.$$

Then $\{f_{\vec{r}} : \vec{r} \in (\mathbb{F}_q)^\ell\}$ is an SU $(q^\ell; 2^\ell - 1, q)$-hash family.

In the constructions we have presented so far, the hash functions are all linear functions. A construction for an SU hash family of quadratic functions was presented in [8].

Theorem 3.4 *Let q be an odd prime power. For $a, b \in \mathbb{F}_q$, define $f_{a,b} : \mathbb{F}_q \to \mathbb{F}_q$ by the rule*

$$f_{a,b}(x) = (x + a)^2 + b.$$

Then $\{f_{a,b} : a, b \in \mathbb{F}_q\}$ is an SU $(q^2; q, q)$-hash family.

Proof. Clearly we have a $(q^2; q, q)$-hash family. We prove that it is SU: Let $x_1, x_2 \in \mathbb{F}_q$ (where $x_1 \neq x_2$) and let $y_1, y_2 \in \mathbb{F}_q$. We want to show that the number of ordered pairs $(a, b) \in (\mathbb{F}_q)^2$ such that

$$(x_1 + a)^2 + b = y_1$$

and

$$(x_2 + a)^2 + b = y_2$$

is a constant. Subtracting the two equations, we can solve uniquely for a:

$$a = \frac{y_1 - y_2}{2(x_1 - x_2)} - \frac{x_1 + x_2}{2}.$$

Then, given a, we obtain a unique solution for b. □

It is interesting to note that the hash family constructed in Theorem 3.4 is also equivalent to the desarguesian affine plane of order q. The SU $(9; 3, 3)$ hash family presented in Example 2.2 is an application of Theorem 3.4.

In view of Theorem 2.1, we can construct many δ-U hash families from codes. For example, using Reed-Solomon codes, we obtain the following construction which was pointed out in [3].

Theorem 3.5 *Let q be a prime power and let $1 \leq k \leq q-1$. For $a \in \mathbb{F}_q$, define $f_a : (\mathbb{F}_q)^k \rightarrow \mathbb{F}_q$ by the rule*

$$f_a(x_0, \ldots, x_{k-1}) = x_0 + \sum_{i=1}^{k-1} x_i a^i.$$

Then $\{f_a : a \in \mathbb{F}_q\}$ is a $\frac{k-1}{q}$-U $(q; q^k, q)$ hash family.

Proof. Clearly we have a $(q; q^k, q)$-hash family. We prove that it is $\frac{k-1}{q}$-U: Let

$$(x_0, \ldots, x_{k-1}), (x'_0, \ldots, x'_{k-1}) \in (\mathbb{F}_q)^k$$

be two different vectors. We want to determine (an upper bound on the) the number of elements $a \in \mathbb{F}_q$ such that

$$\sum_{i=0}^{k-1} x_i a^i = \sum_{i=0}^{k-1} x'_i a^i.$$

This is equivalent to

$$\sum_{i=0}^{k-1} (x_i - x'_i) a^i = 0.$$

Since a non-zero polynomial of degree at most $k-1$ over a field $\mathbb{F}_q$ has at most $k-1$ roots, it follows that there are at most $k-1$ values of a for which this equation holds. Therefore the hash family is $\frac{k-1}{q}$-U. □

The following construction is given in [21] (it is in fact based on a 1952 construction for difference matrices due to Bose and Bush [4]).

Theorem 3.6 *Let q be a prime power and let s and t be positive integers such that $s \geq t$. Let $\phi : \mathbb{F}_{q^s} \rightarrow (\mathbb{F}_q)^t$ be any surjective q-linear mapping (i.e., $\phi(x+y) = \phi(x) + \phi(y)$ for all $x, y \in \mathbb{F}_{q^s}$, and $\phi(ax) = a\phi(x)$ for all $a \in \mathbb{F}_q$, $x \in \mathbb{F}_{q^s}$). For every $a \in \mathbb{F}_{q^s}$, define $f_a : \mathbb{F}_{q^s} \rightarrow (\mathbb{F}_q)^t$ by the rule*

$$f_a(x) = \phi(ax).$$

Then $\{f_a : a \in \mathbb{F}_{q^s}\}$ is a $\frac{1}{q^t}$-U $(q^s; q^s, q^t)$ hash family.

Proof. Clearly we have a $(q^s; q^s, q^t)$-hash family. We prove that it is $\frac{1}{q^t}$-U: Let $x_1, x_2 \in \mathbb{F}_{q^s}$, $x_1 \neq x_2$. We want to determine (an upper bound on the) the number of elements $a \in \mathbb{F}_q$ such that

$$\phi(ax_1) = \phi(ax_2).$$

Since ϕ is linear, this is equivalent to

$$\phi(a(x_1 - x_2)) = 0.$$

Now, since ϕ is surjective and linear, we have that $|\ker(\phi)| = q^{s-t}$. Since $x_1 - x_2 \neq 0$, there are exactly q^{s-t} values of a such that $a(x_1 - x_2) \in \ker(\phi)$. Therefore the hash family is $\frac{1}{q^t}$-U, as desired. □

It is possible to take the composition of two hash families whenever the domain of the functions in one family is the same as the range of the functions in the other family. We now recall a composition construction from [20] which allows us to compose a δ_1-U hash family with a δ_2-U hash family and obtain a $(\delta_1+\delta_2)$-U hash family. (This procedure can be thought of as constructing a *concatenated code.*)

Theorem 3.7 *Suppose $\mathcal{F}_1$ is a δ_1-U $(D_1; N, M_1)$ hash family of functions from X to Y_1, and $\mathcal{F}_2$ is a δ_2-U $(D_2; M_1, M_2)$ hash family of functions from Y_1 to Y_2. For any $f_1 \in \mathcal{F}_1$, $f_2 \in \mathcal{F}_2$, define $f_1 \circ f_2 : X \to Y_2$ by the rule $f_1 \circ f_2(x) = f_2(f_1(x))$. Then*

$$\{f_1 \circ f_2 : f_1 \in \mathcal{F}_1, f_2 \in \mathcal{F}_2\}$$

is a $(\delta_1 + \delta_2)$-U $(D_1 D_2; N, M_2)$ hash family.

Proof. Fix any two distinct elements $x, x' \in X$. We want to compute an upper bound on the number of pairs (f_1, f_2) such that $f_2(f_1(x)) = f_2(f_1(x'))$. Let $\mathcal{G} = \{f_1 \in \mathcal{F}_1 : f_1(x) = f_1(x')\}$. Clearly $|\mathcal{G}| \leq \delta_1 D_1$, and for any $f_1 \in \mathcal{G}$, it holds that $f_2(f_1(x)) = f_2(f_1(x'))$ for all $f_2 \in \mathcal{F}_2$.

Now, if $f_1 \in \mathcal{F}_1 \backslash \mathcal{G}$, then $f_1(x) \neq f_1(x')$. For every $f_1 \in \mathcal{F}_1 \backslash \mathcal{G}$, it follows that there are at most $\delta_2 D_2$ functions $f_2 \in \mathcal{F}_2$ such that $f_2(f_1(x)) = f_2(f_1(x'))$. Therefore the total number of pairs (f_1, f_2) such that $f_2(f_1(x)) = f_2(f_1(x'))$ is at most

$$\begin{aligned} |\mathcal{G}|D_2 + (D_1 - |\mathcal{G}|)\delta_2 D_2 &\leq |\mathcal{G}|D_2 + D_1\delta_2 D_2 \\ &\leq \delta_1 D_1 D_2 + D_1 \delta_2 D_2 \\ &= (\delta_1 + \delta_2) D_1 D_2. \end{aligned}$$

Therefore the hash family is $(\delta_1 + \delta_2)$-U. □

4 Three Applications

In this section, we present three interesting applications of universal hash families.

4.1 Quasirandom Number Generation

Our first application uses strongly universal hash families for quasirandom number generation. Suppose that $\mathcal{F}$ is an SU $(D; N, M)$-hash family of functions from X to Y. A particular function $f \in \mathcal{F}$ is chosen randomly and kept secret. Then one or more values $x \in X$ are chosen using a specified probability distribution p on X. For each x that is chosen, the value $y = f(x)$ is computed and outputted. The objective is that resulting distribution of the values y that are outputted is "close to uniform". Using the theory that we develop later in the paper, it can be shown that the strongly universal property implies that this objective is met for most choices of $f \in \mathcal{F}$, provided that the parameters D, N and M are chosen appropriately and the probability distribution p is reasonably close to uniform.

The BPV generator (see [5]) provides a nice example of the above-described technique. This generator uses Corollary 3.3 as follows. Let p be prime. The set X consists of all $2^\ell - 1$ non-zero binary ℓ-tuples. Now, a vector $\vec{r} \in (\mathbb{Z}_p)^\ell$ is chosen at random. This determines a fixed function $f_{\vec{r}} \in \mathcal{F}(p, \ell, X)$. Then random choices of $\vec{x}$ are made, and for each $\vec{x}$ that is selected, the value $f_{\vec{r}}(\vec{x})$ is computed and outputted.

The BPV generator is useful because it allows a convenient method of precomputation to speed up random number generation. This could be useful in the context of implementing signature schemes on a constrained device such as a smart card. Suppose we want to implement an ElGamal-type signature scheme which requires computing a pair (y, α^y) for a secret random value y, where α generates a cyclic subgroup of a finite field $\mathbb{F}_p$, for some prime, p. We might use the BPV generator in order to generate the required y-values, i.e., $y = f_{\vec{r}}(\vec{x}) = \vec{r} \cdot \vec{x}$ where $\vec{x} \in \{0,1\}^\ell$ is chosen randomly and $\vec{r} = (r_1, \ldots, r_\ell) \in (\mathbb{F}_p)^\ell$ is fixed. If we precompute the values $\alpha^{r_1}, \ldots, \alpha^{r_\ell}$ and store them, then

$$\alpha^y = \prod_{\{i : x_i = 1\}} \alpha^{r_i}$$

can be computed by multiplying together a subset of these ℓ precomputed values. This replaces an exponentiation modulo p by (at most) $\ell - 1$ multiplications in $\mathbb{F}_p$.

4.2 Derandomization

A BPP (bounded-error probabilistic polynomial time) algorithm is a randomized algorithm, say A, for a decision problem, that returns the correct answer ("yes" or "no") for any possible problem instance I with probability at least 3/4. The algorithm A chooses a random value $y \in Y$, where Y is a specified finite set, and then proceeds deterministically, computing an output that is denoted $\mathsf{A}(I, y)$. The algorithm A should have the property that, for any instance I, $\mathsf{A}(I, y)$ yields the correct answer for at least $3|Y|/4$ of the values $y \in Y$.

Suppose that A uses m random bits to choose the random value $y \in Y$ (so $|Y| = 2^m$). Then we can decrease the probability of error as follows:

1. Choose k values $y_1, \ldots, y_k \in Y$ uniformly at random;
2. Run $\mathsf{A}(I, y_i)$ for $1 \leq i \leq k$; and
3. Take the majority output ("yes" or "no") as the final answer.

Using the Chernoff bound, it is not hard to show that the error probability is reduced to $2^{-\Omega(k)}$ by this method; however, we require km random bits to apply this technique.

Deterministic amplification is any method of partial derandomization of a probabilistic algorithm. The goal is to reduce the error probability as much as possible, while requiring fewer random bits than the method described above. (In general, there will be a tradeoff between the number of random bits used and the error probability.)

The following is a useful technique for deterministic amplification: Suppose that $\mathcal{F}$ is a δ-U $(D; N, M)$-hash family of functions from X to Y, where $M = 2^m$. Further, as described above, suppose that we have a randomized algorithm, say A, in the class BPP, in which each run of the algorithm depends on a random value chosen from the set Y. Let I be any problem instance, and consider the following algorithm:

1. Choose a random element $x \in X$;
2. Run the algorithm $\mathsf{A}(I, f(x))$ for all $f \in \mathcal{F}$;
3. Take the majority output ("yes" or "no") as the final answer.

We show in §8 that the resulting error probability can be bounded as a function of the error probability of A by using "extraction" properties of δ-U $(D; N, M)$-hash families. Similar techniques can be used for deterministic amplification of RP (randomized polynomial) algorithms, which are probabilistic algorithms having (bounded) one-sided error. (For a survey on these topics, see [16].)

4.3 Privacy Amplification

The concept of *privacy amplification* is due to Bennett, Brassard and Robert ([2]). Suppose that two parties, Alice and Bob, can carry out a key agreement protocol using quantum cryptography, at the end of which they each know the value of some element $x \in X$. An eavesdropper, Eve, has some partial information on the value of x, which is specified by a probability distribution p on X. Alice and Bob do not know the probability distribution p but they have some information about the non-uniformity of the distribution p, as specified by its collision probability, for example (this concept will be defined in §5.5).

Now, suppose that $\mathcal{F}$ is a δ-U $(D; N, M)$-hash family of functions from X to Y. A particular function $f \in \mathcal{F}$ is chosen randomly by Alice and Bob and kept secret. Alice and Bob can both compute the value $y = f(x)$. The objective is that Eve should have very little information about the value of y. We will show in §9 how Eve's knowledge about y can be bounded suitably, given the parameters D, N and M and a measure of the non-uniformity of p.

5 Statistical Distance and Quasirandomness

Suppose that $\mathcal{F}$ is an SU hash family of functions from X to Y. One main result we will prove is that, with high probability (with respect to the function $f \in \mathcal{F}$), the values $f(x)$ have a close to uniform distribution when $x \in X$ is chosen using a close to uniform distribution. We quantify the notion of "close to uniform distribution" in this section, using the notions of *statistical distance* and *quasirandomness*. We also present some basic definitions and results on topics such as collision probability and entropy of probability distributions.

5.1 Probability Spaces and Random Variables

We begin with a few standard definitions concerning probability spaces and random variables. A *finite probability space* is a pair (Y, p), where Y is a finite set and p is a probability distribution on Y. The *uniform* probability distribution on Y is denoted u_Y; it assigns probability $1/|Y|$ to every element $y \in Y$. A *random variable* on a finite probability space (Y, p) is a function $\mathbf{Y} : Y \to \mathbb{R}$.

The *expectation* of $\mathbf{Y}$, denoted $\mathbf{E}(\mathbf{Y})$, is defined to be

$$\mathbf{E}(\mathbf{Y}) = \sum_{y \in Y} p(y)\mathbf{Y}(y).$$

The *variance* of $\mathbf{Y}$, denoted $\mathbf{var}(\mathbf{Y})$, is defined to be

$$\mathbf{var}(\mathbf{Y}) = \mathbf{E}(\mathbf{Y}^2) - (\mathbf{E}(\mathbf{Y}))^2 = \mathbf{E}((\mathbf{Y} - \mathbf{E}(\mathbf{Y}))^2).$$

If we wish to emphasize the dependence on the particular probability distribution p, we may use the notation $\mathbf{E}_p(\mathbf{Y})$ and $\mathbf{var}_p(\mathbf{Y})$.

We state the following fundamental results from probability theory without proof.

Lemma 5.1 (Chebyshev's inequality) *For any random variable* $\mathbf{Y}$, *it holds that*

$$\mathbf{Pr}[|\mathbf{Y}(y) - \mathbf{E}(\mathbf{Y})| \geq \epsilon] \leq \frac{\mathbf{var}(\mathbf{Y})}{\epsilon^2}.$$

Lemma 5.2 (Jensen's inequality) *Suppose that* $\mathcal{I} \subseteq \mathbb{R}$ *is an interval,* $\mathbf{Y}$ *is a random variable taking on values in* $\mathcal{I}$, *and* $f : \mathcal{I} \to \mathbb{R}$ *is strictly concave on the interval* $\mathcal{I}$. *Then it holds that*

$$\mathbf{E}(f(\mathbf{Y})) \leq f(\mathbf{E}(\mathbf{Y})).$$

By taking $f(x) = -x^2$ and $\mathcal{I} = \mathbb{R}$, we obtain the following corollary:

Corollary 5.3 *For any random variable* $\mathbf{Y}$, *it holds that*

$$(\mathbf{E}(\mathbf{Y}))^2 \leq \mathbf{E}(\mathbf{Y}^2).$$

By taking $f(x) = \log x$ and $\mathcal{I} = (0, \infty)$, we obtain the following corollary:

Corollary 5.4 *For any random variable* $\mathbf{Y}$ *taking on positive values, it holds that*

$$\log \mathbf{E}(\mathbf{Y}) \geq \mathbf{E}(\log \mathbf{Y}).$$

5.2 Statistical Distance

Let p and q be two probability distributions on the set Y. We define the *statistical distance* between p and q, denoted $\mathsf{d}(p, q)$, as follows:

$$\mathsf{d}(p, q) = \frac{1}{2} \sum_{y \in Y} |p(y) - q(y)|.$$

It is easily seen that $0 \leq \mathsf{d}(p, q) \leq 1$ for all probability distributions p and q. Another elementary property is given in the following lemma.

Lemma 5.5 *Let p and q be two probability distributions on the set Y. Then it holds that*

$$\sum_{y \in Y} \max\{p(y), q(y)\} = \mathsf{d}(p, q) + 1.$$

Proof. Let $Y_p = \{y \in Y : p(y) \geq q(y)\}$. Then

$$\begin{aligned}
\mathsf{d}(p, q) &= \frac{1}{2} \sum_{y \in Y_p} (p(y) - q(y)) + \frac{1}{2} \sum_{y \in Y \setminus Y_p} (q(y) - p(y)) \\
&= \frac{1}{2} \sum_{y \in Y_p} p(y) - \frac{1}{2} \sum_{y \in Y \setminus Y_p} p(y) - \frac{1}{2} \sum_{y \in Y_p} q(y) + \frac{1}{2} \sum_{y \in Y \setminus Y_p} q(y) \\
&= \frac{1}{2} \sum_{y \in Y_p} p(y) - \frac{1}{2} \left(1 - \sum_{y \in Y_p} p(y) \right) - \frac{1}{2} \sum_{y \in Y_p} q(y) \\
&\qquad + \frac{1}{2} \left(1 - \sum_{y \in Y_p} q(y) \right) \\
&= \sum_{y \in Y_p} p(y) - \sum_{y \in Y_p} q(y).
\end{aligned}$$

However, we also have that

$$\begin{aligned}
\sum_{y \in Y} \max\{p(y), q(y)\} &= \sum_{y \in Y_p} p(y) + \sum_{y \in Y \setminus Y_p} q(y) \\
&= \sum_{y \in Y_p} p(y) + 1 - \sum_{y \in Y_p} q(y).
\end{aligned}$$

It therefore follows that

$$\sum_{y \in Y} \max\{p(y), q(y)\} = \mathsf{d}(p, q) + 1.$$

□

Let p be any probability distribution on the set Y. For any $Y_0 \subseteq Y$, define

$$p(Y_0) = \sum_{y \in Y_0} p(y).$$

Lemma 5.6 *Let p and q be two probability distributions on the set Y. Then it holds that*

$$\mathsf{d}(p, q) = \max\{|p(Y_0) - q(Y_0)| : Y_0 \subseteq Y\}.$$

Proof. Define Y_p as in the proof of Lemma 5.5. Note that

$$|p(Y_p) - q(Y_p)| = \sum_{y \in Y_p} (p(y) - q(y)).$$

We showed in the proof of Lemma 5.5 that

$$\sum_{y \in Y_p} (p(y) - q(y)) = \mathsf{d}(p, q).$$

Therefore, $|p(Y_p) - q(Y_p)| = \mathsf{d}(p, q)$. It is also easy to see that

$$q(Y \backslash Y_p) - p(Y \backslash Y_p)| = \mathsf{d}(p, q).$$

To complete the proof, we show that $|p(Y_p) - q(Y_p)| \geq |p(Y_0) - q(Y_0)|$ for all $Y_0 \subseteq Y$. Let $Y_0 \subseteq Y$, and denote $Y_1 = Y_0 \cap Y_p$ and $Y_2 = Y_0 \cap (Y \backslash Y_p)$. Observe that $p(Y_1) - q(Y_1) > 0$ and $p(Y_2) - q(Y_2) < 0$. Then we have that

$$\begin{aligned} p(Y_0) - q(Y_0) &= p(Y_1) - q(Y_1) + (p(Y_2) - q(Y_2)) \\ &\leq p(Y_1) - q(Y_1) \\ &\leq p(Y_p) - q(Y_p) \qquad \text{since } Y_1 \subseteq Y_p \\ &= |p(Y_p) - q(Y_p)|. \end{aligned}$$

Similarly,

$$\begin{aligned} q(Y_0) - p(Y_0) &= q(Y_2) - p(Y_2) + (q(Y_1) - p(Y_1)) \\ &\leq q(Y_2) - p(Y_2) \\ &\leq q(Y \backslash Y_p) - p(Y \backslash Y_p) \qquad \text{since } Y_2 \subseteq Y \backslash Y_p \\ &= |p(Y_p) - q(Y_p)|. \end{aligned}$$

Therefore we have that

$$|p(Y_0) - q(Y_0)| \leq |p(Y_p) - q(Y_p)|,$$

as desired. □

Example 5.1 Consider the following two probability distributions p and q on the set $\{y_1, y_2, y_3, y_4\}$:

	$p(y_i)$	$q(y_i)$
y_1	1/3	1/4
y_2	1/3	1/4
y_3	1/6	1/4
y_4	1/6	1/4

We can compute $\mathsf{d}(p,q)$ by any one of the three methods described above. Using the definition of distance, we compute

$$\mathsf{d}(p,q) = \frac{1}{2} \times 4 \times \frac{1}{12} = \frac{1}{6}.$$

If we use Lemma 5.5, then we compute

$$\mathsf{d}(p,q) = 2 \times \frac{1}{3} + 2 \times \frac{1}{4} - 1 = \frac{1}{6}.$$

Finally, using Lemma 5.6, we have

$$\mathsf{d}(p,q) = p(\{y_1, y_2\}) - q(\{y_1, y_2\}) = \frac{2}{3} - \frac{1}{2} = \frac{1}{6}.$$

We get the same answer in each case, of course! ∎

5.3 Distinguishability

Statistical distance of probability distributions is related to the concept of distinguishability. Let p_0 and p_1 be two probability distributions on the set Y. Consider the probability distribution q defined on the set Y by choosing $i \in \{0,1\}$ uniformly at random, and then choosing $y \in Y$ with probability $p_i(y)$. It is easy to see that

$$q(y) = \frac{p_0(y) + p_1(y)}{2}.$$

A *distinguisher* is a function $f : Y \to \{0,1\}$. Intuitively, given a value $y \in Y$ chosen by the above method (i.e., according to the probability distribution q), the distinguisher is trying to guess whether it is more likely that $i = 0$ or $i = 1$. Suppose we denote by $corr(f)$ the probability that the distinguisher f makes a correct guess for i, given y. The probability that the distinguisher f is correct, given $y \in Y$, is

$$\frac{p_{f(y)}(y)}{p_0(y) + p_1(y)}.$$

From this, it follows immediately that

$$corr(f) = \sum_{y \in Y} \frac{p_0(y) + p_1(y)}{2} \times \frac{p_{f(y)}(y)}{p_0(y) + p_1(y)} = \sum_{y \in Y} \frac{p_{f(y)}(y)}{2}.$$

For each $y \in Y$, a distinguisher will maximize its probability of guessing the value of i correctly by choosing $i \in \{0,1\}$ so that

$$p_i(y) \geq p_{1-i}(y).$$

Therefore, the *optimal distinguisher*, denoted f^*, is defined as follows:

$$f^*(y) = \begin{cases} 0 & \text{if } p_1(y) < p_0(y) \\ 1 & \text{if } p_1(y) \geq p_0(y). \end{cases}$$

The probability that the distinguisher f^* is correct is

$$corr(f^*) = \sum_{y \in Y} \frac{\max\{p_0(y), p_1(y)\}}{2} = \frac{\mathsf{d}(p_0, p_1) + 1}{2},$$

where the last equality follows from Lemma 5.5.

We can view f and f^* as random variables on both of the probability spaces (Y, p_0) and (Y, p_1). The following easily proven relation compares the statistical distance of p_0 and p_1 to the expectation of f and f^* over the two probability spaces.

Lemma 5.7 *Suppose that p_0 and p_1 are probability distributions defined on a set Y. Then, for any $f : Y \rightarrow \{0, 1\}$, it holds that*

$$|\mathbf{E}(f_{p_1}) - \mathbf{E}(f_{p_0})| \leq \mathbf{E}(f^*_{p_1}) - \mathbf{E}(f^*_{p_0}) = \mathsf{d}(p_0, p_1).$$

Proof. We first show that $\mathbf{E}(f^*_{p_1}) - \mathbf{E}(f^*_{p_0}) = \mathsf{d}(p_0, p_1)$. We compute

$$\begin{aligned} \mathbf{E}(f^*_{p_1}) &= \sum_{y \in Y} f^*(y) p_1(y) \\ &= \sum_{\{y \in Y : p_1(y) \geq p_0(y)\}} p_1(y). \end{aligned}$$

Similarly,

$$\mathbf{E}(f^*_{p_0}) = \sum_{\{y \in Y : p_1(y) \geq p_0(y)\}} p_0(y).$$

Then it follows that $\mathbf{E}(f^*_{p_1}) - \mathbf{E}(f^*_{p_0}) = \mathsf{d}(p_0, p_1)$, as in the proof of Lemma 5.5.

We complete the proof by showing that $|\mathbf{E}(f_{p_1}) - \mathbf{E}(f_{p_0})| \leq \mathsf{d}(p_0, p_1)$. This is seen as follows:

$$|\mathbf{E}(f_{p_1}) - \mathbf{E}(f_{p_0})| = |p_1(f^{-1}(1)) - p_0(f^{-1}(1))| \leq \mathsf{d}(p_0, p_1),$$

where the inequality follows from Lemma 5.6. □

5.4 Quasirandomness

Let (Y, p) be a finite probability space, let $Y_0 \subseteq Y$, and let $\epsilon > 0$ be a real number Then we say that p is *quasirandom within ϵ with respect to Y_0* provided that

$$\left| p(Y_0) - \frac{|Y_0|}{|Y|} \right| \leq \epsilon.$$

Further, p is *quasirandom within ϵ* provided that

$$\left| p(Y_0) - \frac{|Y_0|}{|Y|} \right| \leq \epsilon$$

for all $Y_0 \subseteq Y$.

Using the fact that $u_Y(Y_0) = |Y_0|/|Y|$, the following is an immediate corollary of Lemma 5.6.

Lemma 5.8 *Let u_Y be the uniform probability distribution on the set Y. Then an arbitrary probability distribution p on the set Y is quasirandom within ϵ if and only if $\mathsf{d}(p, u_Y) \leq \epsilon$.*

5.5 Collision Probability

Let (Y, p) be a probability space. The *collision probability* of the probability distribution p is defined to be the quantity

$$\Delta_p = \sum_{y \in Y} (p(y))^2.$$

Observe that $\Delta_p = 1/|Y|$ if $p = u_Y$. We now prove a relationship between collision probability and quasirandomness of a probability distribution that is due to Impagliazzo and Zuckerman [13, Claim 2].

Lemma 5.9 *Let (Y, p) be a probability space. Then p is quasirandom within $\sqrt{\Delta_p |Y| - 1}/2$.*

Proof. Let $|Y| = M$. Using the fact that $\Delta_p = \sum (p(y))^2$, it follows that

$$\sum_{y \in Y} \left(p(y) - \frac{1}{M} \right)^2 = \Delta_p - \frac{1}{M}.$$

Let $\mathbf{Y}$ be the random variable on the probability space (Y, u_Y) defined by the formula $\mathbf{Y}(y) = |p(y) - (1/M)|$. Then

$$\mathbf{E}(\mathbf{Y}^2) = \frac{1}{M} \left(\Delta_p - \frac{1}{M} \right) = \frac{\Delta_p M - 1}{M^2}.$$

Applying Corollary 5.3, we have that

$$\mathbf{E}(\mathbf{Y}) \leq \sqrt{\mathbf{E}(\mathbf{Y}^2)} = \frac{\sqrt{\Delta_p M - 1}}{M}.$$

Now we compute

$$\mathsf{d}(p, u_Y) = \frac{1}{2} \sum_{y \in Y} \left| p(y) - \frac{1}{M} \right| = \frac{M}{2} \times \mathbf{E}(\mathbf{Y}) \leq \frac{\sqrt{\Delta_p M - 1}}{2}.$$

□

5.6 Shannon, Renyi and Min Entropy

Let (Y, p) be a probability space. The *Renyi entropy* of (Y, p), denoted $h_{\mathsf{Ren}}(p)$, is defined to be

$$h_{\mathsf{Ren}}(p) = -\log_2 \Delta_p.$$

The *min entropy* of (Y, p), denoted $h_{\mathsf{min}}(p)$, is defined to be

$$h_{\mathsf{min}}(p) = \min\{-\log_2 p(y) : y \in Y\} = -\log_2(\max\{p(y) : y \in Y\}).$$

The *Shannon entropy* of (Y, p), denoted $h_(p)$, is defined to be

$$h(p) = -\sum_{y \in Y} p(y) \log_2 p(y).$$

Observe that the uniform distribution u_Y has

$$h(u_Y) = h_{\mathsf{Ren}}(u_Y) = h_{\mathsf{min}}(u_Y) = \log_2 |Y|.$$

The following lemma is easy to prove.

Lemma 5.10 *Let (Y, p) be a probability space. Then $h_{\mathsf{Ren}}(p)/2 \leq h_{\mathsf{min}}(p) \leq h_{\mathsf{Ren}}(p) \leq h(p)$.*

Proof. First, we have that

$$(\max\{p(y) : y \in Y\})^2 \leq \sum (p(y))^2.$$

This implies that $h_{\mathsf{Ren}}(p)/2 \leq h_{\mathsf{min}}(p)$.

Next, we observe that

$$\sum (p(y))^2 \leq \sum (p(y) \times \max\{p(y) : y \in Y\}) = \max\{p(y) : y \in Y\}.$$

It therefore follows that $h_{\mathsf{min}}(p) \leq h_{\mathsf{Ren}}(p)$.

Finally, we define the random variable $\mathbf{Y}$ on the probability space (Y, p) by the rule $\mathbf{Y}(y) = p(y)$. Note that $E(\mathbf{Y}) = \sum(p(y))^2$ and $E(\log \mathbf{Y}) = \sum p(y) \log p(y)$. Now apply Corollary 5.4, obtaining the following:

$$\log \left(\sum (p(y))^2 \right) \geq \sum p(y) \log p(y).$$

It therefore follows that $h_{\mathsf{Ren}}(p) \leq h(p)$. □

Several results in the literature on the applications described in §4 involve probability distributions with specified bounds on their min or Renyi entropy. The significance of the above lemma is that these two quantities differ by a factor of two at most, so they can be used essentially interchangably (up to a constant factor). We will generally state our results in terms of collision probability in this paper.

6 Quasirandomness of SU Hash Families

In this section, we will state and prove some results regarding quasirandomness of SU hash families. We provide a somewhat simpler treatment of several theorems from [17]. Our approach is similar to that used in [9, §B.2].

Suppose that (X, p) is a finite probability space and $\mathcal{F}$ is any SU $(D; N, M)$-hash family of functions from X to Y. For any $f \in \mathcal{F}$, define the induced probability distribution on q_f on Y as follows:

$$q_f(y) = \sum_{x \in f^{-1}(y)} p(x)$$

for all $y \in Y$. $q_f(y)$ is the probability that the output of the function f takes on the value y, given that $x \in X$ is chosen using the probability distribution p.

For any $y \in Y$, we define a random variable χ_y on the probability space $(\mathcal{F}, u_{\mathcal{F}})$ as follows:

$$\chi_y(f) = q_f(y)$$

for all $f \in \mathcal{F}$. It is easy to see that

$$\sum_{f \in \mathcal{F}} \chi_y(f) = \frac{D}{M}.$$

Hence, we have that

$$\mathbf{E}(\chi_y) = \frac{1}{M}. \tag{1}$$

We now prove the following important combinatorial lemma.

Lemma 6.1 *Suppose that (X,p) is a finite probability space and $\mathcal{F}$ is any* SU $(D;N,M)$*-hash family of functions from X to Y. Let $y \in Y$, and let χ_y be the random variable on $\mathcal{F}$ that was defined above. Then it holds that*

$$\sum_{f\in\mathcal{F}}(\chi_y(f))^2 = \frac{D(1+(M-1)\Delta_p)}{M^2}.$$

Proof.

$$\begin{aligned}
\sum_{f\in\mathcal{F}}(\chi_y(f))^2 &= \sum_{f\in\mathcal{F}}\left(\sum_{x\in f^{-1}(y)} p(x)\right)^2 \\
&= \sum_{f\in\mathcal{F}}\sum_{x_1\in f^{-1}(y)}\sum_{x_2\in f^{-1}(y),x_2\neq x_1} p(x_1)p(x_2) \\
&\quad + \sum_{f\in\mathcal{F}}\sum_{x\in f^{-1}(y)}(p(x))^2 \\
&= \frac{D}{M^2}\sum_{x_1\in X}\sum_{x_2\in X,x_2\neq x_1} p(x_1)p(x_2) + \frac{D}{M}\sum_{x\in X}(p(x))^2 \\
&= \left(\frac{D}{M^2}\right)(1-\Delta_p) + \left(\frac{D}{M}\right)\Delta_p \\
&= \frac{D(1+(M-1)\Delta_p)}{M^2}.
\end{aligned}$$

□

Example 6.1 We present an SU $(9;3,3)$-hash family, with a particular probability distribution p imposed on X. In the last column of the following table, we record the values of χ_0:

	$p(0)=1/2$	$p(1)=1/4$	$p(2)=1/4$	χ_0
$f_{0,0}$	0	1	1	$\frac{1}{2}$
$f_{0,1}$	1	2	2	0
$f_{0,2}$	2	0	0	$\frac{1}{2}$
$f_{1,0}$	1	1	0	$\frac{1}{4}$
$f_{1,1}$	2	2	1	0
$f_{1,2}$	0	0	2	$\frac{3}{4}$
$f_{2,0}$	1	0	1	$\frac{1}{4}$
$f_{2,1}$	2	1	2	0
$f_{2,2}$	0	2	0	$\frac{3}{4}$

Then we have

$$\sum(\chi_0(f_{a,b}))^2 = \frac{7}{4} = \frac{9\left(1+(3-1)\left(\left(\frac{1}{2}\right)^2+\left(\frac{1}{4}\right)^2+\left(\frac{1}{4}\right)^2\right)\right)}{3^2},$$

as was shown in Lemma 6.1. ∎

As an immediate corollary of Lemma 6.1, we have that

$$\mathbf{E}(\chi_y{}^2) = \frac{1+(M-1)\Delta_p}{M^2}. \tag{2}$$

Now, using Equations (1) and (2), we obtain the following:

$$\mathbf{var}(\chi_y) = \mathbf{E}(\chi_y{}^2) - (\mathbf{E}(\chi_y))^2 = \frac{(M-1)\Delta_p}{M^2}.$$

Then, applying Chebyshev's inequality (Lemma 5.1), we have that

$$\mathbf{Pr}[|\chi_y(f) - \mathbf{E}(\chi_y)| \geq \epsilon] \leq \frac{(M-1)\Delta_p}{\epsilon^2 M^2}.$$

Finally, observe that

$$|\chi_y(f) - \mathbf{E}(\chi_y)| = \left|q_f(y) - \frac{1}{M}\right|.$$

Therefore q_f is quasirandom with respect to y if and only if

$$|\chi_y(f) - \mathbf{E}(\chi_y)| \leq \epsilon,$$

and we have proven the following, which is a slight generalization of [9, Lemma B.3] and [17, Theorem 5].

Theorem 6.2 *Suppose that (X,p) is a finite probability space and $\mathcal{F}$ is any* SU *$(D;N,M)$-hash family of functions from X to Y. Let $y \in Y$ be fixed, and let $f \in \mathcal{F}$ be chosen randomly. Then the probability that q_f is not quasirandom within ϵ with respect to y is at most*

$$\frac{(M-1)\Delta_p}{\epsilon^2 M^2}.$$

The preceding result can be generalized to handle the situation of quasirandomness with respect to any $Y_0 \subseteq Y$. The following can be proven easily in a similar fashion.

Theorem 6.3 *Suppose that (X, p) is a finite probability space and $\mathcal{F}$ is any* SU $(D; N, M)$*-hash family of functions from X to Y. Let $Y_0 \subseteq Y$ be fixed, and let $f \in \mathcal{F}$ be chosen randomly. Then the probability that q_f is not quasirandom within ϵ with respect to Y_0 is at most*

$$\frac{|Y_0|(M - |Y_0|)\Delta_p}{\epsilon^2 M^2}.$$

Theorem 6.4 *Suppose that (X, p) is a finite probability space and $\mathcal{F}$ is any* SU $(D; N, M)$*-hash family of functions from X to Y. Let $f \in \mathcal{F}$ be chosen randomly. Then the probability that q_f is not quasirandom within ϵ is at most*

$$\frac{\Delta_p M(M-1)}{4\epsilon^2.}$$

Proof. First, we observe that, if

$$\left| q_f(y) - \frac{1}{M} \right| < \frac{2\epsilon}{M}$$

for all $y \in Y$, then $\mathsf{d}(q_f, u_Y) \leq \epsilon$, and hence q_f will be quasirandom within ϵ by Lemma 5.8.

Let $f \in \mathcal{F}$ be chosen at random. For any $y \in Y$, the probability that

$$\left| q_f(y) - \frac{1}{M} \right| > \frac{2\epsilon}{M}$$

is at most

$$\frac{\Delta_p(M-1)}{(2\epsilon/M)^2 M^2} = \frac{\Delta_p(M-1)}{4\epsilon^2}.$$

Since there are M choices for $y \in Y$, the probability that $\left|q_f(y) - \frac{1}{M}\right| > 2\epsilon/M$ for some $y \in Y$ is at most $\Delta_p M(M-1)/(4\epsilon^2)$. Hence, the result follows. □

7 Leftover Hash Lemma

In this section, we discuss and prove a general version of the so-called "leftover hash lemma". An early version of this was proven in [12]; see also [1, 9, 13, 14] for closely related results. The leftover hash lemma concerns the quasirandomness of the probability distribution r on the set $\mathcal{F} \times Y$ defined by choosing $f \in \mathcal{F}$ randomly, and then evaluating $f(x)$ when $x \in X$ is chosen using the probability distribution p. Hence, r is defined as follows:

$$r(f, y) = \frac{q_f(y)}{D} = \frac{\chi_y(f)}{D}.$$

The following lemma establishes a result similar to Lemma 6.1, under the weaker assumption that $\mathcal{F}$ is a δ-U hash family. The proof of Lemma 7.1 is essentially identical to that of Lemma 6.1.

Lemma 7.1 *Suppose that (X,p) is a finite probability space and $\mathcal{F}$ is any δ-U $(D;N,M)$-hash family of functions from X to Y. For all $y \in Y$ and $f \in \mathcal{F}$, let $\chi_y(f) = q_f(y)$. Then it holds that*

$$\sum_{y \in Y} \sum_{f \in \mathcal{F}} (\chi_y(f))^2 \leq D(\delta + (1-\delta)\Delta_p).$$

Example 7.1 We present a $\frac{1}{2}$-U $(4;4,2)$-hash family, with a particular probability distribution p imposed on X. In the last two columns of the following table, we record the values of χ_0 and χ_1:

	$p(0)=1/2$	$p(1)=1/6$	$p(2)=1/6$	$p(3)=1/6$	χ_0	χ_1
f_1	0	0	1	1	$\frac{2}{3}$	$\frac{1}{3}$
f_2	0	0	0	0	1	0
f_3	0	1	1	0	$\frac{2}{3}$	$\frac{1}{3}$
f_4	0	1	0	1	$\frac{2}{3}$	$\frac{1}{3}$

Then we have

$$\Delta_p = \left(\frac{1}{2}\right)^2 + \left(\frac{1}{6}\right)^2 + \left(\frac{1}{6}\right)^2 + \left(\frac{1}{6}\right)^2 = \frac{1}{3}$$

and

$$\sum_{i=0}^{1} \sum_{j=1}^{4} (\chi_i(f_j))^2 = \frac{8}{3} = 4\left(\frac{1}{2} + \left(1-\frac{1}{2}\right)\frac{1}{3}\right),$$

so the bound of Lemma 7.1 is met with equality. ∎

We now state the main result of this section, which follows immediately from Lemma 7.1.

Theorem 7.2 *Suppose that $\mathcal{F}$ is any δ-U $(D;N,M)$-hash family of functions from X to Y. Suppose that p is a probability distribution on X, and let r be the probability distribution that is induced on $\mathcal{F} \times Y$, as defined above. Then*

$$\Delta_r \leq \frac{\delta + (1-\delta)\Delta_p}{D}.$$

Corollary 7.3 *Suppose that $\mathcal{F}$ is any δ-U $(D; N, M)$-hash family of functions from X to Y. Suppose that p is a probability distribution on X, and let r be the probability distribution that is induced on $\mathcal{F} \times Y$, as defined above. Then*

$$\mathsf{d}(u_{\mathcal{F}\times Y}, r) \leq \frac{\sqrt{M(\delta + (1-\delta)\Delta_p) - 1}}{2}.$$

Proof. Apply Lemma 5.9 and Theorem 7.2 . □

8 Extractors

We begin with a definition. Suppose $\mathcal{F}$ is a δ-U $(D; N, M)$-hash family of functions from X to Y. Let p be a probability distribution on X, and let the probability distribution r be defined as in §7. We say that $\mathcal{F}$ is a (k, ϵ)-*extractor* if $\mathsf{d}(u_{\mathcal{F}\times Y}, r) < \epsilon$ whenever $h_{\mathsf{Ren}}(p) \geq k$. The following result provides a sufficient condition for a given hash family to be an extractor.

Theorem 8.1 *A δ-U $(D; N, M)$-hash family is a (k, ϵ)-extractor if*

$$\sqrt{M(\delta + 2^{-k}) - 1} \leq 2\epsilon$$

Proof. Apply Corollary 7.3 with $\Delta_p = 2^{-k}$. Then

$$\mathsf{d}(u_{\mathcal{F}\times Y}, r) \leq \frac{\sqrt{M(\delta + (1-\delta)2^{-k}) - 1}}{2} < \frac{\sqrt{M(\delta + 2^{-k}) - 1}}{2} \leq \epsilon.$$

□

We now prove that a $(k, 1/4)$-extractor allows the the error probability of a BPP algorithm to be reduced to $2^k/N$, using the technique described in §4.2. Suppose that $\mathcal{F}$ is a δ-U $(D; N, M)$-hash family of functions from X to Y. Further, suppose that we have a randomized algorithm, say A, in the class BPP, in which each run of the algorithm depends on a random value chosen from the set Y. Let I be any problem instance; choose a random element $x \in X$; and run the algorithm $\mathsf{A}(I, f(x))$ for all $f \in \mathcal{F}$. Define $\mathsf{B}(I, x)$ to be the the majority output ("yes" or "no"). The following result, concerning the error probability of the algorithm B, is from [16].

Theorem 8.2 *The error probability of the algorithm* B*, as described above, is at most $2^k/N$.*

Proof. Let I be any problem instance. Let $Y_0 \subseteq Y$ consist of all y-values such that $\mathsf{A}(I, y)$ yields the wrong answer. Then $|Y_0| \leq |Y|/4 = M/4$. For any $x \in X$, define $B_x = \{f \in \mathcal{F} : f(x) \in Y_0\}$. B_x consists of all functions

f such that $\mathsf{A}(I, f(x))$ returns the wrong answer. Observe that $\mathsf{B}(I, x)$ returns the wrong answer if and only if $|B_x| \geq D/2$. Define $X_0 = \{x \in X : |B_x| \geq D/2\}$. The error probability of the algorithm B is $|X_0|/N$. We will complete the proof by showing that $|X_0| \leq 2^k$.

Define a probability distribution p on the set X as follows:

$$p(x) = \begin{cases} \frac{1}{|X_0|} & \text{if } x \in X_0 \\ 0 & \text{if } x \in X \backslash X_0. \end{cases}$$

It is not hard to prove that the induced probability distribution r satisfies

$$r(\mathcal{F} \times Y_0) \geq \frac{1}{2}.$$

On the other hand, with respect to the uniform distribution on $\mathcal{F} \times Y$, we have that

$$u_{\mathcal{F} \times Y}(\mathcal{F} \times Y_0) = \frac{|Y_0|}{|Y|} \leq \frac{1}{4}.$$

It therefore holds that

$$\mathsf{d}(r, u_{\mathcal{F} \times Y}) \geq \frac{1}{2} - \frac{1}{4} = \frac{1}{4}.$$

Now suppose that $|X_0| > 2^k$. Note that $\Delta_p = 1/|X_0|$, so it follows that $h_{\mathsf{Ren}} > k$. Since our hash family is an extractor with $\epsilon = 1/4$, if follows by definition that

$$\mathsf{d}(r, u_{\mathcal{F} \times Y}) < \frac{1}{4}.$$

This contradiction proves that $|X_0| \leq 2^k$, and the proof is complete. □

It is clear that the number of random bits required by the algorithm B is $\log_2 N$. (These random bits are required to choose a random value $x \in X$.) Also, the number of trials required to run algorithm B (i.e., the number of times that the algorithm A is run during the execution of B) is D.

By using various types of δ-U hash families, we can use Theorem 8.2 to obtain a variety of deterministic amplification techniques simply by plugging the parameters of the hash families into Theorem 8.1. As an illustration, we show that the two-point sampling technique of Chor and Goldreich [7] can be viewed as a special case of this general approach. We let $M = 2^m$, and construct a $\frac{1}{M}$-U $(M; M^2, M)$-hash family using Theorem 3.5 with $k = 2$. Theorem 8.1 says that this hash family is an $(m + 2, 1/4)$-extractor. Therefore the probability amplification result proved in Theorem 8.2 reduces the error probability to $4/M$ using $2 \log_2 M$ random bits and M trials.

In the above example, the error probability is proportional to the reciprocal of the number of trials. In general, it is desired to find a probability amplification technique in which the error probability decreases exponentially quickly as a function of the number of trials, such as is the case when each trial uses independent random bits (as discussed in §4.2). We present a new, simple amplification technique of this type, using hash families that are a slightly modified version of some hash families used for the purpose of unconditionally secure authentication (see [3]).

We require two ingredients. First, applying Theorem 3.5 with $q = 2^{a+m}$ and $k = 2^{a-3}$, we obtain a

$$\tfrac{1}{2^{m+3}}\text{-U } (2^{a+m}; 2^{(a+m)2^{a-3}}, 2^{a+m})$$

hash family. The second ingredient is a

$$\tfrac{1}{2^m}\text{-U } (2^{a+m}; 2^{a+m}, 2^m)$$

hash family, which exists by applying Theorem 3.6 with $q = 2$, $t = m$ and $s = a + m$. Now we compose the two hash families using Theorem 3.7. We get a hash family with parameters

$$\tfrac{9}{8\times 2^m}\text{-U } (2^{2a+2m}; 2^{(a+m)2^{a-3}}, 2^m).$$

Using Theorem 8.1, it is easily verified that this hash family is an $(m + 3, 1/4)$-extractor. Now, if we denote $t = (a + m)2^{a-3} - (m + 3)$, then we can use Theorem 8.2 to reduce the error probability of a BPP algorithm to 2^{-t} using $m + t + 3$ random bits and $2^{2a+2m} = O(t^2)$ trials.

9 Privacy Amplification

In this section, we present some very interesting results on privacy amplification that can be found in [1]. These results can also be viewed as applications of the leftover hash lemma. We use the same scenario as in the two previous sections: $\mathcal{F}$ is a δ-U $(D; N, M)$-hash family of functions from X to Y, p is a probability distribution on X, and the probability distribution r is defined as in the two previous sections. $f \in \mathcal{F}$ is chosen randomly (i.e., using the uniform distribution $u_{\mathcal{F}}$ on $\mathcal{F}$); $x \in X$ is chosen using the probability distribution p; and then $y = f(x)$ is computed. We consider the probability distribution q induced on Y by this process. Clearly the following relations hold:

$$q(y|f) = \chi_y(f) = q_f(y)$$

and

$$q(y) = \sum_{f\in\mathcal{F}} r(f, y) = \sum_{f\in\mathcal{F}} \frac{\chi_y(f)}{D} = \sum_{f\in\mathcal{F}} \frac{q_f(y)}{D}.$$

For any fixed $f \in \mathcal{F}$, q_f is a probability distribution on Y. Therefore we can compute

$$\begin{aligned} h_{\mathrm{Ren}}(q_f) &= -\log_2 \Delta_{q_f} \\ &= -\log_2 \left(\sum_{y \in Y} (q_f(y))^2 \right) \\ &= -\log_2 \left(\sum_{y \in Y} (\chi_y(f))^2 \right). \end{aligned}$$

The Renyi entropy $h_{\mathrm{Ren}}(q|u_{\mathcal{F}})$ is defined to be

$$h_{\mathrm{Ren}}(q|u_{\mathcal{F}}) = \sum_{f \in \mathcal{F}} u_{\mathcal{F}}\, h_{\mathrm{Ren}}(q_f).$$

Hence, we have that

$$h_{\mathrm{Ren}}(q|u_{\mathcal{F}}) = -\sum_{f \in \mathcal{F}} \frac{1}{D} \log_2 \left(\sum_{y \in Y} (\chi_y(f))^2 \right). \tag{3}$$

Applying Corollary 5.4, we have that

$$\sum_{f \in \mathcal{F}} \frac{1}{D} \log_2 \left(\sum_{y \in Y} (\chi_y(f))^2 \right) \leq \log_2 \left(\sum_{f \in \mathcal{F}} \frac{1}{D} \sum_{y \in Y} (\chi_y(f))^2 \right). \tag{4}$$

Now Lemma 7.1 together with equations (3) and (4) imply the following result proved in [1, Theorem 3]:

Theorem 9.1 $h_{\mathrm{Ren}}(q|u_{\mathcal{F}}) \geq -\log_2(\delta + \Delta_p)$.

We continue in the same fashion as [1], by stating and proving a consequence of Theorem 9.1. We return to the setting of privacy amplification introduced in Section 4.3.

Suppose that $x \in X$ is chosen uniformly at random by Alice and Bob, and Eve is given the value of $z = e(x)$, where $e : X \to Z$ is a public *eavesdropping function.* Then Alice and Bob randomly and secretly choose $f \in \mathcal{F}$, and use the value $y = f(x)$ as their secret key.

For each $z \in Z$, define $c_z = |e^{-1}(z)|$. The probability distribution p_z on X, given the value z, is the following:

$$p_z(x) = \begin{cases} \frac{1}{c_z} & \text{if } x \in e^{-1}(z) \\ 0 & \text{if } x \notin e^{-1}(z). \end{cases}$$

Clearly we have $\Delta_{p_z} = 1/c_z$ for all $z \in Z$. Therefore, from Theorem 9.1, it follows that $h_{\mathsf{Ren}}(q|u_{\mathcal{F}}, z) \geq -\log_2(\delta + 1/c_z)$. (The notation $h_{\mathsf{Ren}}(q|u_{\mathcal{F}}, z)$ means that the value z is fixed.)

Now we compute the average Renyi entropy $h_{\mathsf{Ren}}(q|u_{\mathcal{F}}, z)$ over all possible values of z. Since $x \in X$ is shosen randomly and $z = e(x)$, this average should be computed using the probability distribution e_Z on Z that is defined as $e_Z(z) = c_z/|X| = c_z/N$ for all $z \in Z$. Now we can compute this average entropy to be

$$\begin{aligned}
h_{\mathsf{Ren}}(q|u_{\mathcal{F}}, e_Z) &= \sum_{z \in Z} e_Z(z)\, h_{\mathsf{Ren}}(q|u_{\mathcal{F}}, z) \\
&\geq -\sum_{z \in Z} \frac{c_z}{N} \log_2 \left(\delta + \frac{1}{c_z}\right) \\
&\geq -\log_2 \left(\sum_{z \in Z} \frac{c_z}{N} \left(\delta + \frac{1}{c_z}\right)\right) \qquad \text{from Corollary 5.4} \\
&= \log_2 N - \log_2 \left(\sum_{z \in Z} (1 + \delta c_z)\right) \\
&= \log_2 N - \log_2 \left(|Z| + \delta N\right).
\end{aligned}$$

Here is an interpretation of this result: Eve's average (Shannon) information about the key (i.e., the value of y), given z, is $h(q) - h(q|u_{\mathcal{F}}, e_Z)$, where h denotes Shannon entropy. By a fundamental propoerty of Shannon entropy, it holds that $h(q) \leq \log_2 |Y| = \log_2 M$. As well, it holds that

$$h(q|u_{\mathcal{F}}, e_Z) \geq h_{\mathsf{Ren}}(q|u_{\mathcal{F}}, e_Z).$$

Using the bound on $h_{\mathsf{Ren}}(q|u_{\mathcal{F}}, e_Z)$ proven above, we obtain the following result proven in [1, Corollary 5].

Theorem 9.2 *Eve's average information about the key $y = f(x)$, given the value $z = e(x)$, is at most*

$$\log_2 M - \log_2 N + \log_2 \left(|Z| + \delta N\right).$$

10 Remarks and Conclusion

We have discussed several variants of the leftover hash lemma and its applications in cryptography and complexity. We have tried to point out the fundamental combinatorial nature of the lemma, and the similarity of the various applications, all of which make use of basic inqualities from probability theory such as Jensen's and Chebyshev's inequalites. We have

also emphasized the close connections between universal hash families, error correcting codes and orthogonal arrays. These links allow the easy derivation of many useful classes of hash families via well-known results from coding theory and combinatorial design theory (and we presented an interesting new class of extractors using this approach). It is our belief that coding theory is "right" way to view hash families and that the enormous amount of research on coding theory in the last 50 or so years has not been exploited to its full potential in the study of universal hash families and their many applications.

Acknowledgements

D.R. Stinson's research is supported by NSERC grants IRC #216431-96 and RGPIN #203114-98.

References

[1] C.H. Bennett, G. Brassard, C. Crépeau and U. Maurer. Generalized privacy amplification. *IEEE Transactions on Information Theory* **41** (1995), 1915–1923.

[2] C.H. Bennett, G. Brassard and J-M. Robert. Privacy amplification by public discussion. *SIAM Journal on Computing* **17** (1988), 210–229.

[3] J. Bierbrauer, T. Johansson, G. Kabatianskii and B. Smeets. On families of hash functions via geometric codes and concatenation. *Lecture Notes in Computer Science* **773** (1994), 331–342 (CRYPTO '93).

[4] R.C. Bose and K.A. Bush. Orthogonal arrays of strength two and three. *Annals Math. Statistics* **23** (1952), 508–524.

[5] V. Boyko, M. Peinado and R. Venkatesan. Speeding up discrete log and factoring based schemes via precomputation. *Lecture Notes in Computer Science* **1403** (1998), 221–235 (EUROCRYPT '98).

[6] J.L. Carter and M.N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences* **18** (1979), 143–154.

[7] B. Chor and O. Goldreich. On the power of two-point based sampling. *Journal of Complexity* **5** (1989), 96–106.

[8] M. Etzel, S. Patel and Z. Ramzan. Square hash: fast message authenticaiton via optimized universal hash functions, *Lecture Notes in Computer Science* **1666** (1999), 234–251 (CRYPTO '99).

[9] O. Goldreich. *Modern Cryptography, Probabilistic Proofs and Pseudorandomness.* Springer-Verlag, 1999.

[10] K. Gopalakrishnan and D.R. Stinson. A simple analysis of the error probability of two-point based sampling. *Information Processing Letters* **60** (1996), 91–96.

[11] A.S. Hedayat, N.J.A. Sloane and J. Stufken. *Orthogonal Arrays: Theory and Applications.* Springer-Verlag, 1999.

[12] R. Impagliazzo, L. Levin and M. Luby. Pseudo-random generation from one-way functions. In *21st ACM Symposium on Theory of Computing*, 1989, pp. 12–24.

[13] R. Impagliazzo and D. Zuckerman. How to recycle random bits. In *30th IEEE Symposium on Foundations of Computer Science*, 1989, pp. 248–253.

[14] M. Luby. *Pseudorandomness and Cryptographic Applications.* Princeton University Press, 1996.

[15] F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error-correcting Codes.* North-Holland, 1977.

[16] N. Nisan and A. Ta-Shma. Extracting randomness: a survey and new constructions. *J. Comput. System Sci.* **58** (1999), 148–173.

[17] P. Nguyen and J. Stern. The hardness of the hidden subset sum problem and its cryptographic application. *Lecture Notes in Computer Science* **1666** (1999), 31–46 (CRYPTO '99).

[18] C.R. Rao. Factorial experiments derivable from combinatorial arrangements of arrays. *Journal of the Royal Statistical Society* **9** (1947), 128–139.

[19] D.R. Stinson. Combinatorial techniques for universal hashing. *Journal of Computer and System Sciences* **48** (1994), 337–346.

[20] D.R. Stinson. Universal hashing and authentication codes. *Designs, Codes and Cryptography* **4** (1994), 369–380.

[21] D.R. Stinson. On the connections between universal hashing, combinatorial designs and error-correcting codes. *Congressus Numerantium* **114** (1996), 7–27.

[22] M.N. Wegman and J.L. Carter. New hash functions and their use in authentication and set equality. *Journal of Computer and System Sciences* **22** (1981), 265–279.