Entropy Module (revised)

SAND2020-10855PE

Bill Cordwell
Mark Torgerson
Sandia National Labs

29 SEP 2020

cordwell@sandia.gov
mdtorge@sandia.gov

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S.Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.
SAND No. 20XX-XXXX.

Entropy Outline

Motivation and Different Entropy Measures

Collisions & Entropy – Random Functions

Yes/No Questions

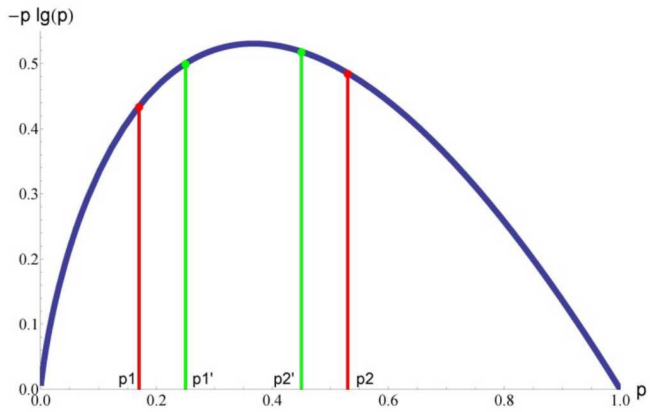
Min-Entropy, Guessing Entropy

Mutual Information

Entropy for keys

PUF Discussion

Fuzzy Extraction



Entropy – “Explanation”

We shall see that the mathematical definition of entropy is relatively straightforward

Understanding what it means, including how and when to apply the definition, is less straightforward

From Wikipedia
In information theory, the entropy of a random variable is the average level of "information", "surprise", or "uncertainty" inherent in the variable’s possible outcomes.

Entropy - definitions

Rényi Entropy

$$H_\alpha = \frac{1}{1-\alpha} \cdot lg\left(\sum_{i=1}^{n} p_i^\alpha\right)$$

Collision Entropy

$$H_2 = -lg\left(\sum_{i=1}^{n} p_i^2\right)$$

Shannon Entropy

$$H_1 = -\sum_{i=1}^{n} p_i \cdot lg(p_i)$$

Min-Entropy

$$H_\infty = -lg\left(\max_{i=1,\ldots,n}(p_i)\right)$$

Exercise – lg(x)

$$\lg(x) = \frac{\log(x)}{\log(2)} = \frac{\ln(x)}{\ln(2)}$$

Find lg(4) by using the log(.) key

0.60206 / 0.30103 = 2

Find lg(4) by using the ln(.) key

1.38629 / 0.693147 = 2

What is lg(4)? What power of 2?

Calculation – fair coin flip

What is the Shannon entropy of a flip of a fair coin?

Shannon Entropy

$$H_1 = -\sum_{i=1}^{n} p_i \cdot lg(p_i)$$

p_1 = ½ , probability of getting a "head"
p_2 = ½ , probability of getting a "tail"

$$H_1 = -(p_1 \cdot lg(p_1) + p_2 \cdot lg(p_2))$$

$$= -\left[\frac{1}{2} \cdot lg\left(\frac{1}{2}\right) + \frac{1}{2} \cdot lg\left(\frac{1}{2}\right)\right]$$

$$= -\left[\frac{1}{2} \cdot (-1) + \frac{1}{2} \cdot (-1)\right] = \frac{1}{2} + \frac{1}{2} = 1 \text{ (bit)}$$

Biased coin flip

What is the Shannon entropy of a biased coin flip?

p_1 = 3/4 , probability of getting a "head"
p_2 = 1/4 , probability of getting a "tail"

Shannon Entropy

$$H_1 = -\sum_{i=1}^{n} p_i \cdot lg(p_i)$$

$$H_1 = -\left[\frac{3}{4} \cdot lg\left(\frac{3}{4}\right) + \frac{1}{4} \cdot lg\left(\frac{1}{4}\right)\right]$$

$$= -\left[0.75 \cdot lg(0.75) + 0.25 \cdot (-2)\right]$$

$$= 0.8113 \quad (1 \text{ flip})$$

$$lg(x) = \frac{log(x)}{log(2)}$$

Biased coin flip

What is the Min entropy of a biased coin flip?

p_1 = 3/4 , probability of getting a "head"
p_2 = 1/4 , probability of getting a "tail"

Min-Entropy

$$\mathsf{H}_\infty = -lg\left(\max_{i=1,\ldots,n} (p_i) \right)$$

$$\mathsf{H}_\infty = -lg\left[\left(\frac{3}{4}\right)\right]$$

$$\approx 0.415$$

Calculation – Uniform distribution

Uniform distribution, equal probability $p_i = 1/N$

$N = 2^n$ (n bits)

Shannon Entropy

$$H_1 = -\sum_{i=1}^{N} p_i \cdot lg(p_i)$$

$$H_1 = -\sum_{i=1}^{N} \frac{1}{N} \cdot lg\left(\frac{1}{N}\right)$$

$$= -N \cdot \frac{1}{N} \cdot lg\left(\frac{1}{N}\right)$$

$$= lg(N) = lg(2^n) = n$$

Min-Entropy

$$H_\infty = -lg(\max_{i=1,\ldots,n}(p_i))$$

$$H_\infty = -lg(\max(p_i)))$$

$$= -lg\left(\frac{1}{N}\right)$$

$$= lg(N) = lg(2^n) = n$$

Combination Lock

Combination lock – three numbers, all between 0 and 99

If chosen at random, $100^3 = 1{,}000{,}000$
$H_1 = 19.93$

There are 15,222 six-letter words, currently in the English language, April 2010

If a word is chosen at random (some might give the same combination),
$H_1 = 13.9$

Common six-letter words
1,500?
$H_1 = 10.55$

Calculation – Passwords

Upper case letter, 26
Lower case letter, 26
Digits, 10
Special Characters, 10

Require 8 characters
no repetitions

$H_1 = 48.78$ ($\lg(72 \cdot 71 \cdots 65)$)

Require 8 characters
no repetitions
at least 1 upper, 1 lower, 1 digit, 1 special

$H_1 = 47.67$

Calculation - Permutations

Permutation of N objects

N! possibilities, assume equally likely

What is the Shannon entropy?

$$H_1 = -\sum_{i=1}^{N!} \frac{1}{N!} \cdot lg\left(\frac{1}{N!}\right)$$

$$= -N! \cdot \frac{1}{N!} \cdot lg\left(\frac{1}{N!}\right)$$

$$= lg(N!) \approx N \cdot lg(N) - lg(e) \cdot N$$

N!	$lg(N!)$
25!	83.682
29!	102.802
31!	112.663
35!	132.924
47!	197.365
58!	260.343
65!	302.018

Ordering values—a Permutation

Let $n_1, n_2, \ldots, n_N$ be nodes with values

Order the nodes

$$n_{37} < n_6 < n_{24} < n_{13} < n_1 < n_{40} < \ \ldots < n_{19}$$

This *is* a permutation of $n_1, n_2, \ldots, n_N$

If all permutations are equally likely, the entropy is

$$H_1 = \lg(N!)$$

Permutations—a caution

- For $1 \leq i < j \leq N$ define
  - $b_{\mathrm{i,j}} = 1$ if $n_i < n_j$ or $b_{\mathrm{i,j}} = 0$ if $n_i > n_j$
- There are $\frac{N(N-1)}{2} \approx \frac{N^2}{2}$ bits defined this way

Can't we make a $\frac{N^2}{2}$ bit key with $\frac{N^2}{2}$ bits of entropy?

N	$lg(N!)$	$N^2/2$
25	83.7	3[illegible]5
35	132.9	[illegible]
65	302.0	[illegible]112.5

Data Representation

- Suppose $A = \{0,1,2,\ldots,31\}$
- Suppose $B = \{(0,0,0,0,0),\ldots,(1,1,1,1,1)\}$

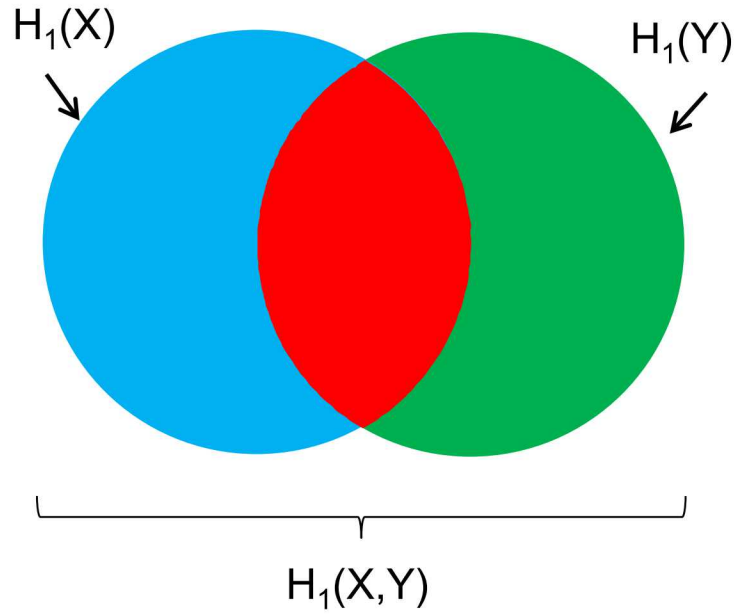
Is there a difference between the two from an entropy standpoint?

- You must know the probability of each event.
  - The way the data is represented does not play a role in the entropy calculation

- However, in B there are other considerations

Multi-dimensions

$$H_1(X,Y) = -\sum_{i=1}^{n} p_i(x,y) \cdot lg(p_i(x,y))$$



How do you compute probability in multiple dimensions?

Independence

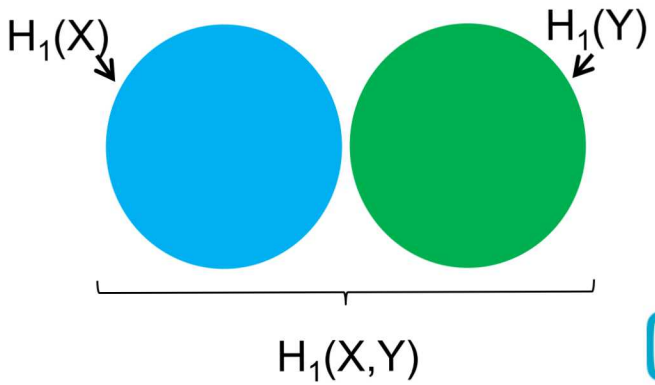
If x and y are independent: $p_i(x,y) = p_a(x) \cdot p_b(y) \Rightarrow$

$$H_1(X,Y) = -\sum_{a=1}^{n_x}\sum_{b=1}^{n_y} p_a(x) \cdot p_b(y) \cdot (lg(p_a(x)) + lg(p_b(y))$$

$$= -\sum_{a=1}^{n_x} p_a(x) lg(p_a(x)) \sum_{b=1}^{n_y} p_b(y) - \sum_{b=1}^{n_y} p_b(y) lg(p_b(y)) \sum_{a=1}^{n_x} p_a(x)$$

$$= -\sum_{a=1}^{n_x} p_a(x) \cdot (lg(p_a(x))) - \sum_{b=1}^{n_y} p_b(y) \cdot (lg(p_b(y)))$$

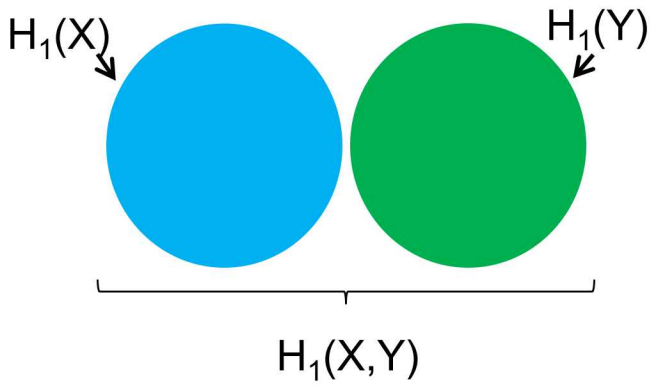
$$H_1(X,Y) = H_1(X) + H_1(Y)$$



Independence

If X and Y are independent:

$$H_1(X,Y) = H_1(X) + H_1(Y)$$



Compute entropy of each dimension, add the results

The same is true of Min entropy, (and the others)

Independence

Compute Shannon entropy of 128 fair coin flips

one flip $H_1 = -\left[\frac{1}{2}\cdot lg\left(\frac{1}{2}\right) + \frac{1}{2}\cdot lg\left(\frac{1}{2}\right)\right] = 1$ x 128 = 128 bits

Compute Shannon entropy of a three number combo lock (0…99) viewed as three independent choices

one spin $H_1 = -lg\left(\frac{1}{100}\right) = 6.64386$ x 3 = 19.9316 bits

Compute Shannon entropy of flipping 128 different coins each with its own probability of success

$$H_1(C_1, \ldots, C_{128}) = \sum_{i=1}^{128} H_1(C_i)$$

Independence

Compute Min entropy of 128 biased coin flips

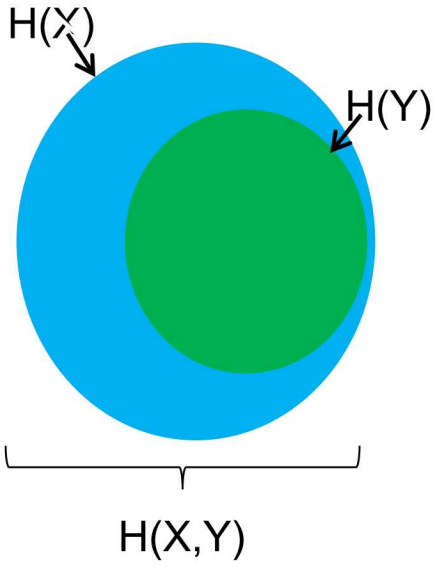
p_1 = 3/4 , probability of getting a "head"
p_2 = 1/4 , probability of getting a "tail"

What is the most likely value? *All heads: p = $(3/4)^{128}$*

$$\mathsf{H}_\infty = -lg\left[\left(\frac{3}{4}\right)^{128}\right]$$
$$= -\left[128 \cdot lg(0.75)\right] \approx 53.1$$

Other Extreme –Fully Dependent

$$H_1(X,Y) = -\sum_{i=1}^{n} p_i(x,y) \cdot lg(p_i(x,y))$$



If y is a function of x, $p_i(x,y) = 0$ unless $y = f(x)$ and then $p_i(x,y) = p_i(x)$

$$\Rightarrow H_1(X,Y) = -\sum_{i=1}^{n} p_i(x,y) \cdot lg(p_i(x,y))$$

$$= \sum_{i=1}^{n_x} p_i(x,f(x)) \cdot lg(p_i(x,f(x)))$$

$$= \sum_{i=1}^{n_x} p_i(x) \cdot lg(p_i(x)) = H_1(X)$$

$$H_1(X,Y) = H_1(X)$$

The same is true of Min-entropy (and the others)

Fully Dependent

- Let $X = \{0,1,\ldots,255\}$ where each element is equally likely
  - $H_1(X) = 8$
- Let $Y = \{0,1,\ldots,8\}$ the set of Hamming weights of X

The joint space (X, Y) has a natural probability distribution

- With $x \in X, y \in Y$
  - $\Pr(x,y) = \Pr(x)$ if $y = HW(x)$
  - $\Pr(x,y) = 0$ if $y \neq HW(x)$
- $H_1(X,Y) = 8$
- $H_1(Y) =$ $H_1(Y) = -\sum_{i=0}^{8} p_i \cdot lg\,(\mathrm{p_i}) \approx 2.5442$
- $H_\infty(Y) =$ $H_\infty(Y) = -lg\,(70/256) \approx 1.8707$

HW	Probability
0	1/256
1	8/256
2	28/256
3	56/256
4	70/256
5	56/256
6	28/256
7	8/256
8	1/256

Not Independent or Dependent

$$H_1(X,Y) = -\sum_{i=1}^{n} p_i(x,y) \cdot lg(p_i(x,y))$$

How do you determine probability of an event if the dimensions are not independent?

How many measurements do you need to determine if dimensions are independent?

This is where things can get difficult!

Key Points

- If the probability distribution is known, we can calculate the Entropy
- If you can mathematically model the thing or process, you have an advantage
- If you are certain about independence (or know the dependence), you have an advantage
- If your only option is to collect sparse data and estimate, you are at a disadvantage

Conditional Entropy

Given the joint distribution on (X, Y), conditional entropy is defined

- $H_1(X|Y) = -\sum_{x \in X, y \in Y} p(x,y) lg(\frac{p(x,y)}{p(y)})$

From this, a chain rule can be derived

- $H_1(X|Y) = H_1(X,Y) - H_1(Y)$

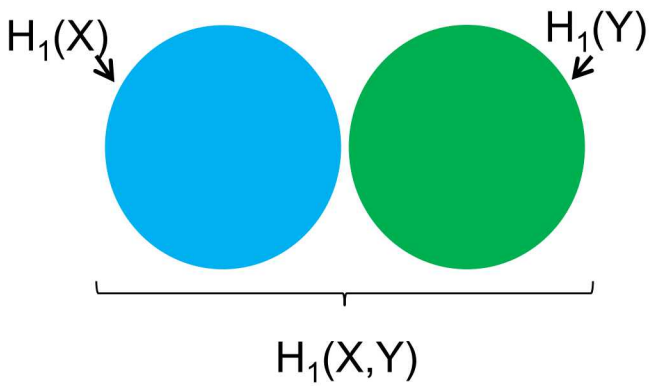
A Bayes type rule follows

- $H_1(Y|X) = H_1(X|Y) + H_1(Y) - H_1(X)$

Independence

If X and Y are independent

- $H_1(X,Y) = H_1(X) + H_1(Y)$



The chain rule gives

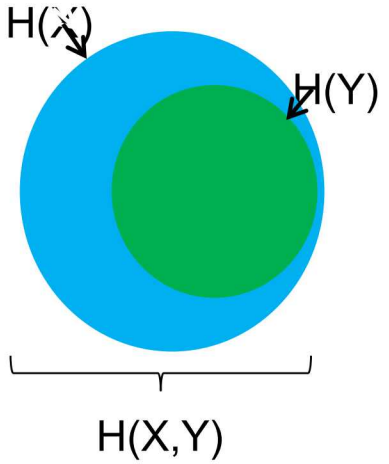
- $H_1(X|Y) = H_1(X,Y) - H_1(Y)$
  $= H_1(X) + H_1(Y) - H_1(Y) = H_1(X)$

When X and Y are independent, revealing Y does not reveal information about X

Fully Dependent

If Y depends on X

- $H_1(X,Y) = H_1(X)$



The chain rule gives

$$H_1(X|Y) = H_1(X,Y) - H_1(Y) = H_1(X) - H_1(Y)$$

Revealing Y *does* reveal information about X.

Hamming Weights

- Let $X = \{0,1, \dots, 255\}$ where each element is equally likely
- Let $Y = \{0,1, \dots, 8\}$ the set of Hamming weights of X

Recall that $H_1(Y) \approx 2.5442$

The chain rule gives

$$H_1(X|Y) = 8 - 2.5442 = 5.4558$$

Revealing the HW of a byte reveals about 2.54 bits of information. (This leaves about 5.46 bits)

Hamming Weights

HW	Probability	Revealed Information	Num bytes	Hidden Info $lg(\#bytes)$	H_{total}
0	$^1/_{256}$	8.0	1	0.0	8.0
1	$^8/_{256}$	5.0	8	3.0	8.0
2	$^{28}/_{256}$	3.19	28	4.81	8.0
3	$^{56}/_{256}$	2.19	56	5.81	8.0
4	$^{70}/_{256}$	1.87	70	6.13	8.0
5	$^{56}/_{256}$	2.19	56	5.81	8.0
6	$^{28}/_{256}$	3.19	28	4.81	8.0
7	$^8/_{256}$	5.0	8	3.0	8.0
8	$^1/_{256}$	8.0	1	0.0	8.0

H_1 = weighted average = 2.54

Larger Hamming Weights

$$H_1 \approx -\frac{1}{2} + \frac{1}{2} \cdot lg(\pi \cdot e \cdot n)$$

H_1 (8 bits) 2.54 bits revealed information

H_1 (16 bits) 3.05 bits revealed information

H_1 (32 bits) 3.55 bits revealed information

H_1 (64 bits) 4.05 bits revealed information

H_1 (128 bits) 4.55 bits revealed information

Error Correction Codes

Suppose that we have a systematic Error Correction Code

- A message M (k bits in length)
- Checkbits $E = ECC(M)$ (e checkbits)
- Codeword $C = M||E$ ($n = k + e$ bits in length)

This is the case of full dependence so

- $H_1(M|E) = H_1(M) - H_1(E) \geq 0$
  - E depends on the messages so $H_1(M) \geq H_1(E)$

For any good code that corrects a lot of errors $e > k$

- $H_1(M) = H_1(E)$ is likely
- $H_1(M|E) = 0$ is likely

Entropy Outline

Motivation and Different Entropy Measures

Collisions & Entropy – Random Functions

Yes/No Questions

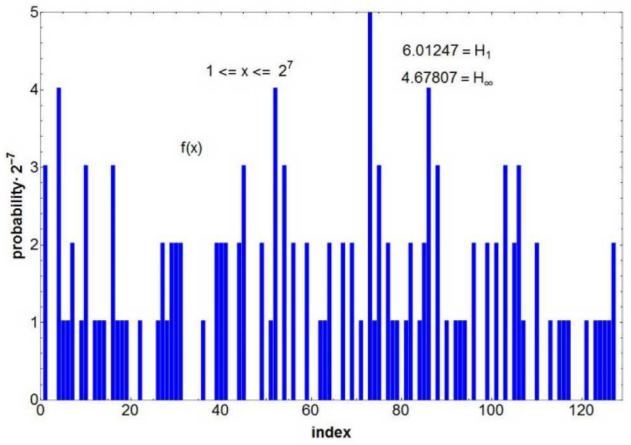
Min-Entropy, Guessing Entropy

Mutual Information

Entropy for keys

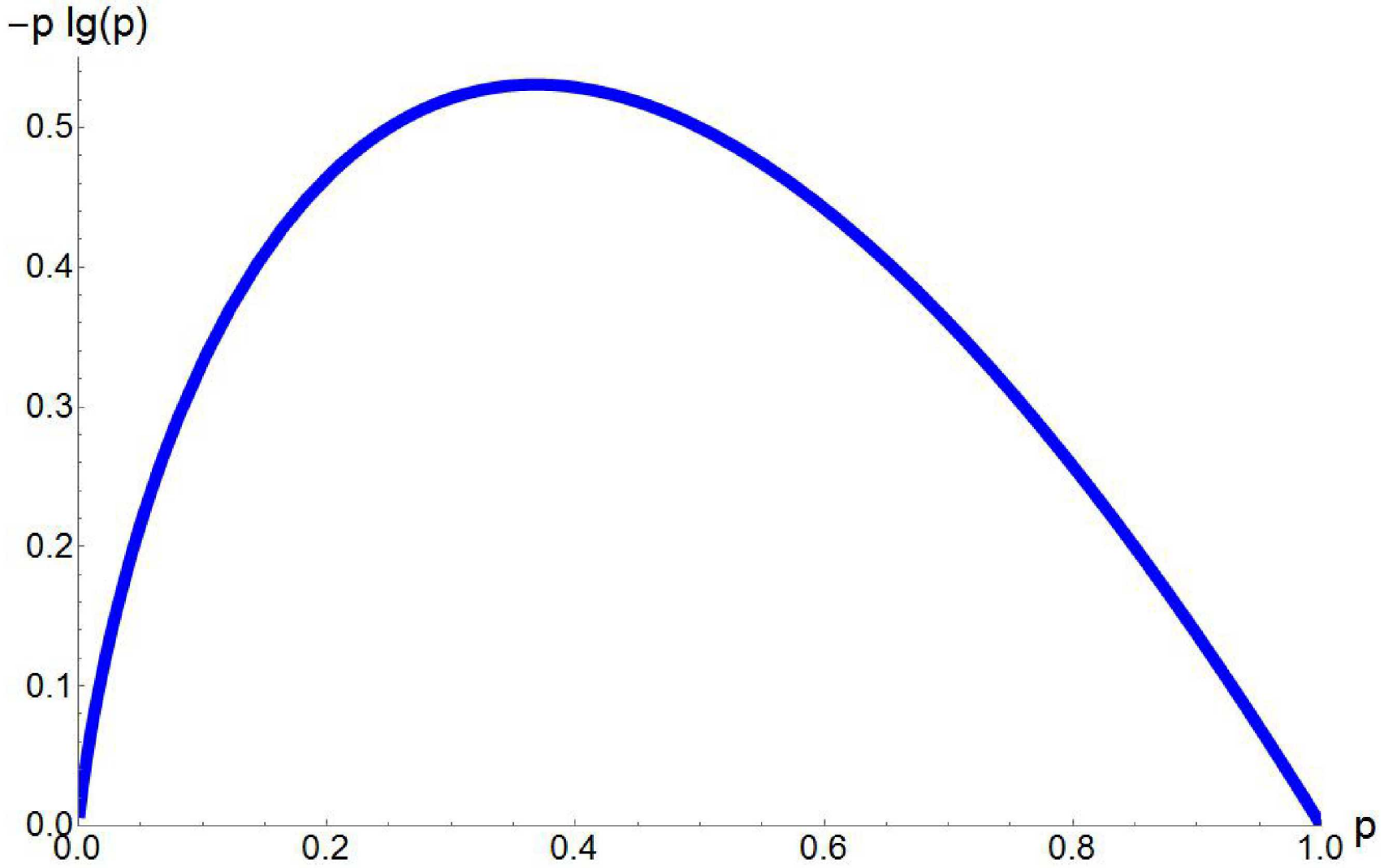
PUF Discussion

Fuzzy Extraction

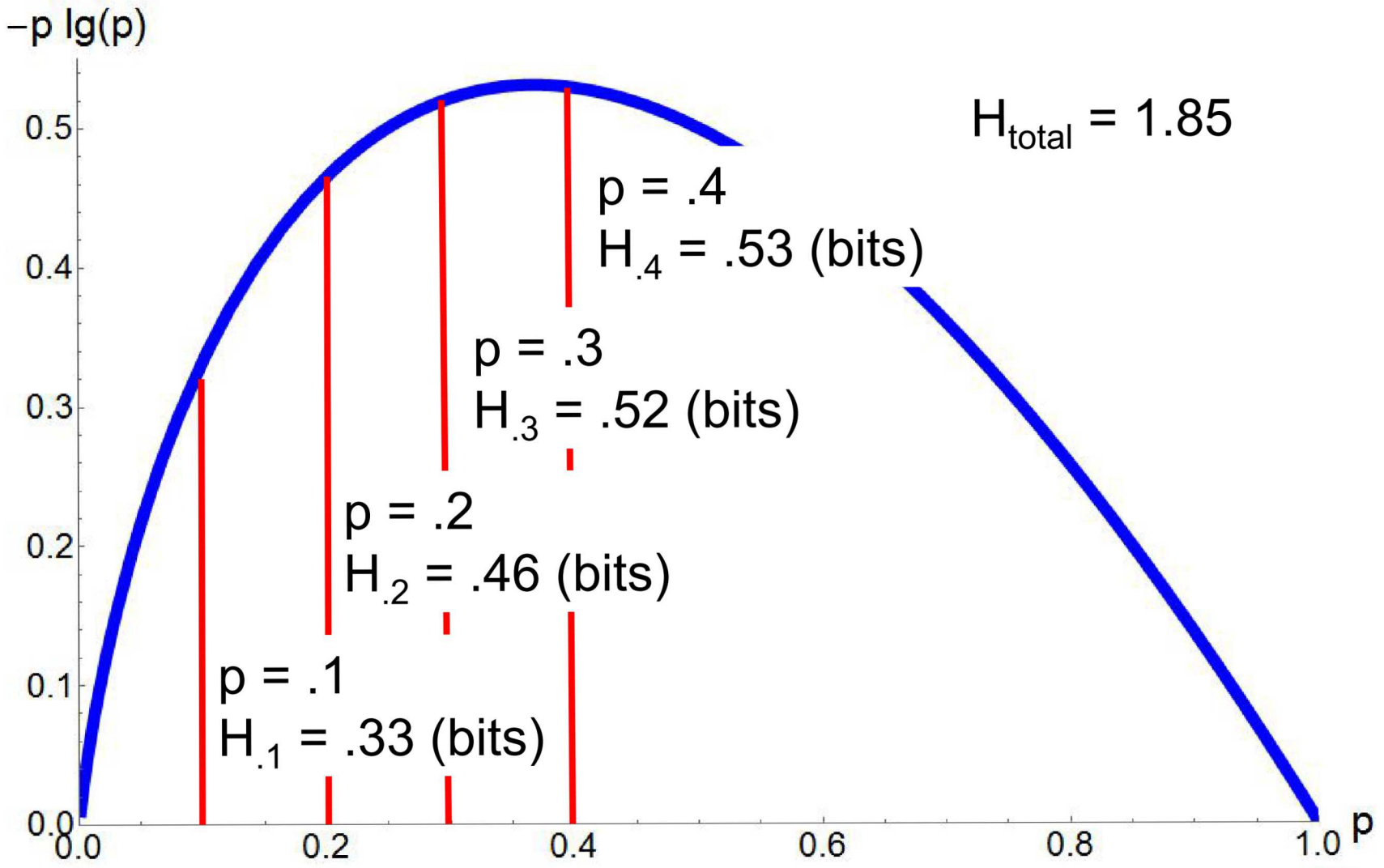


Sandia National Laboratories

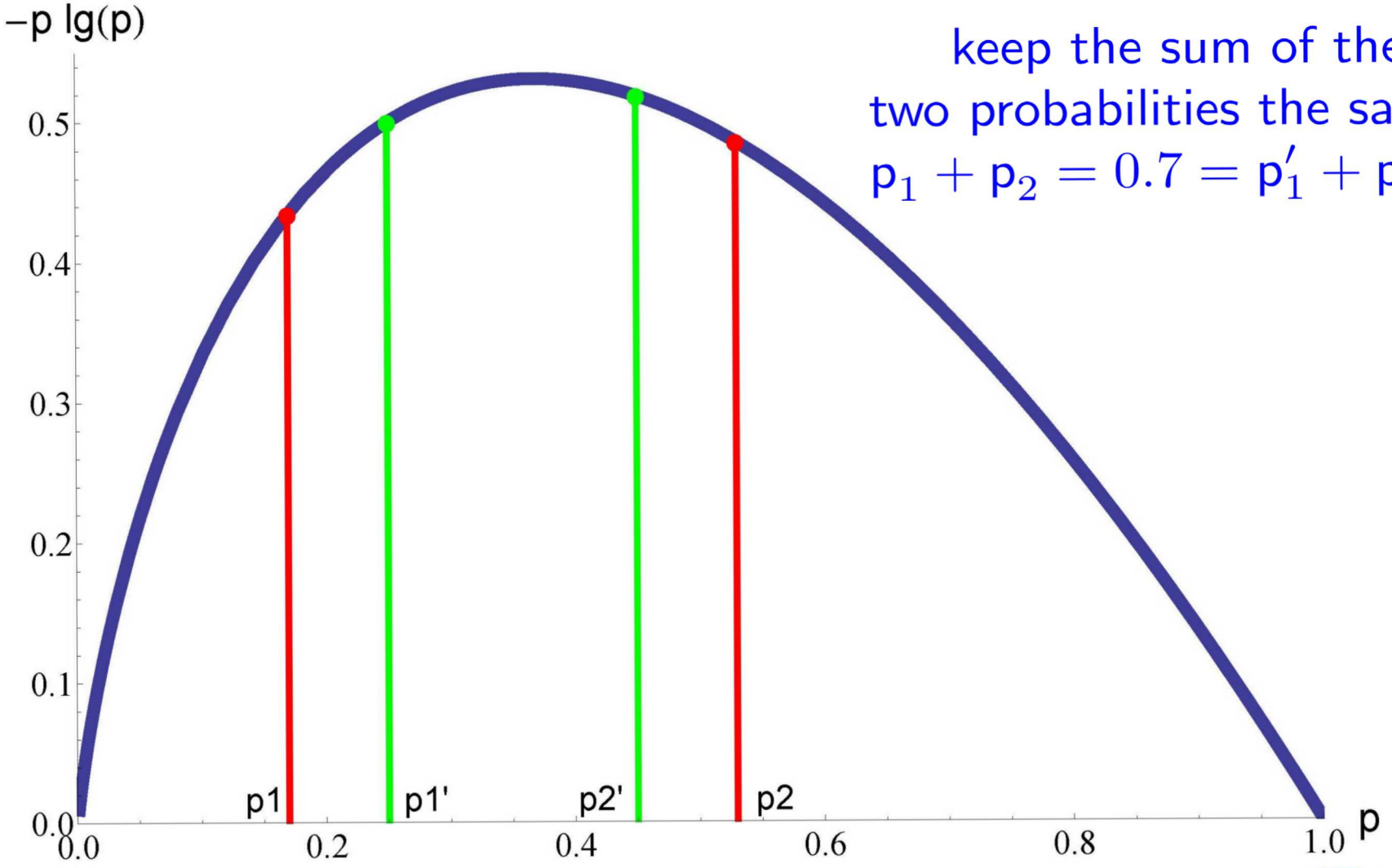
Entropy per point



Entropy per point



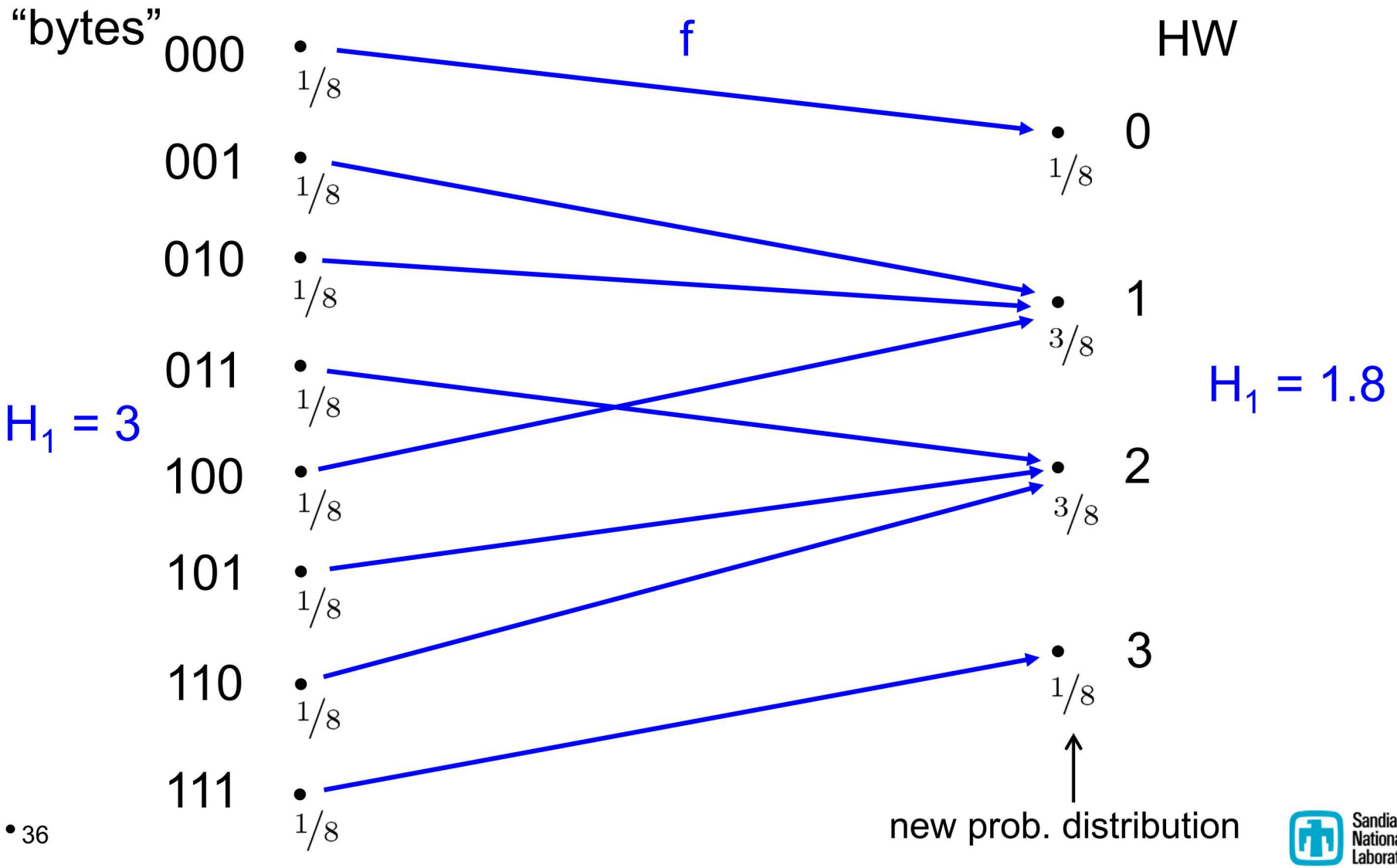
More Uniform $\Rightarrow$ More Entropy



keep the sum of the two probabilities the same

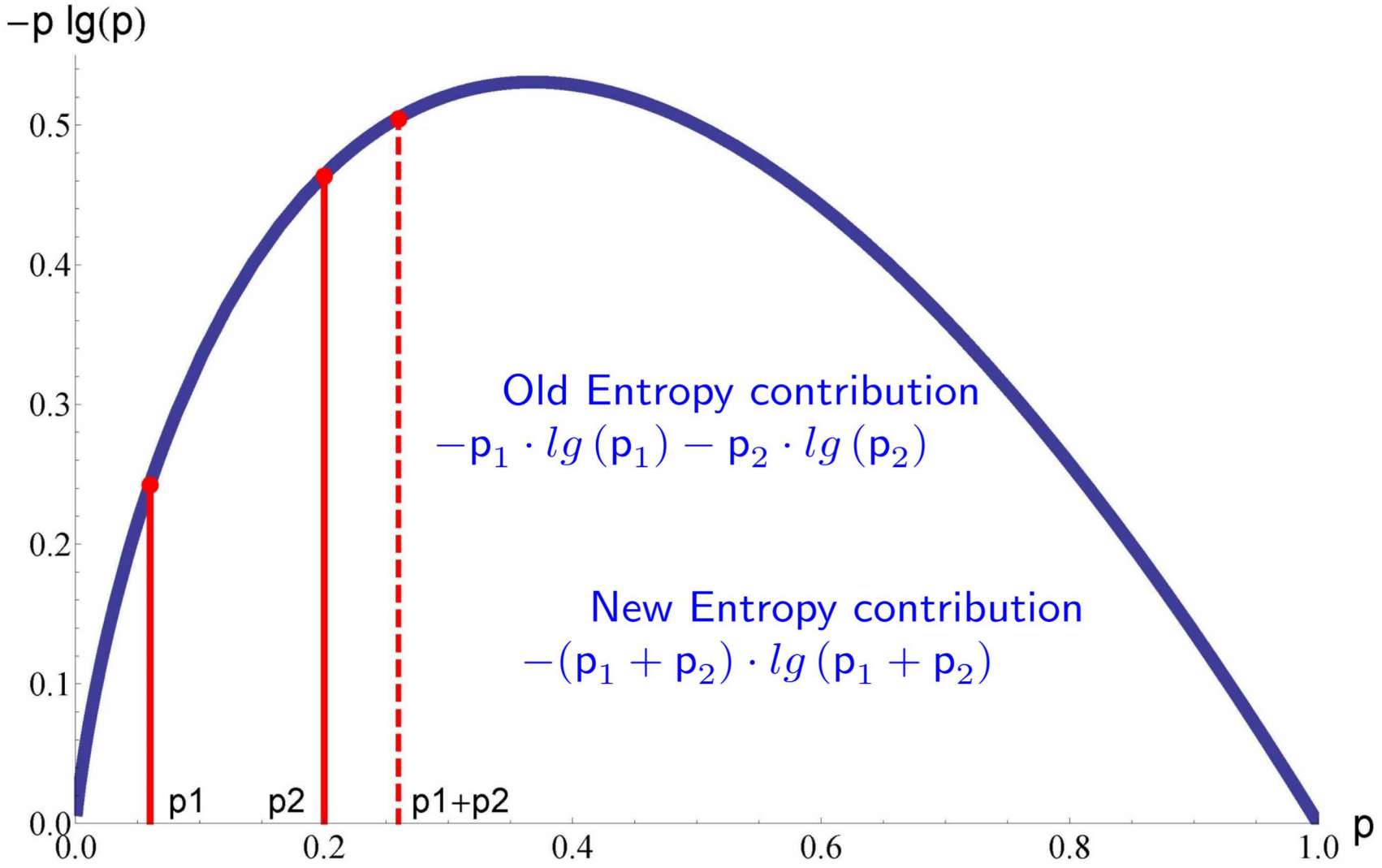
$p_1 + p_2 = 0.7 = p'_1 + p'_2$

Collisions $\Rightarrow$ Less Entropy

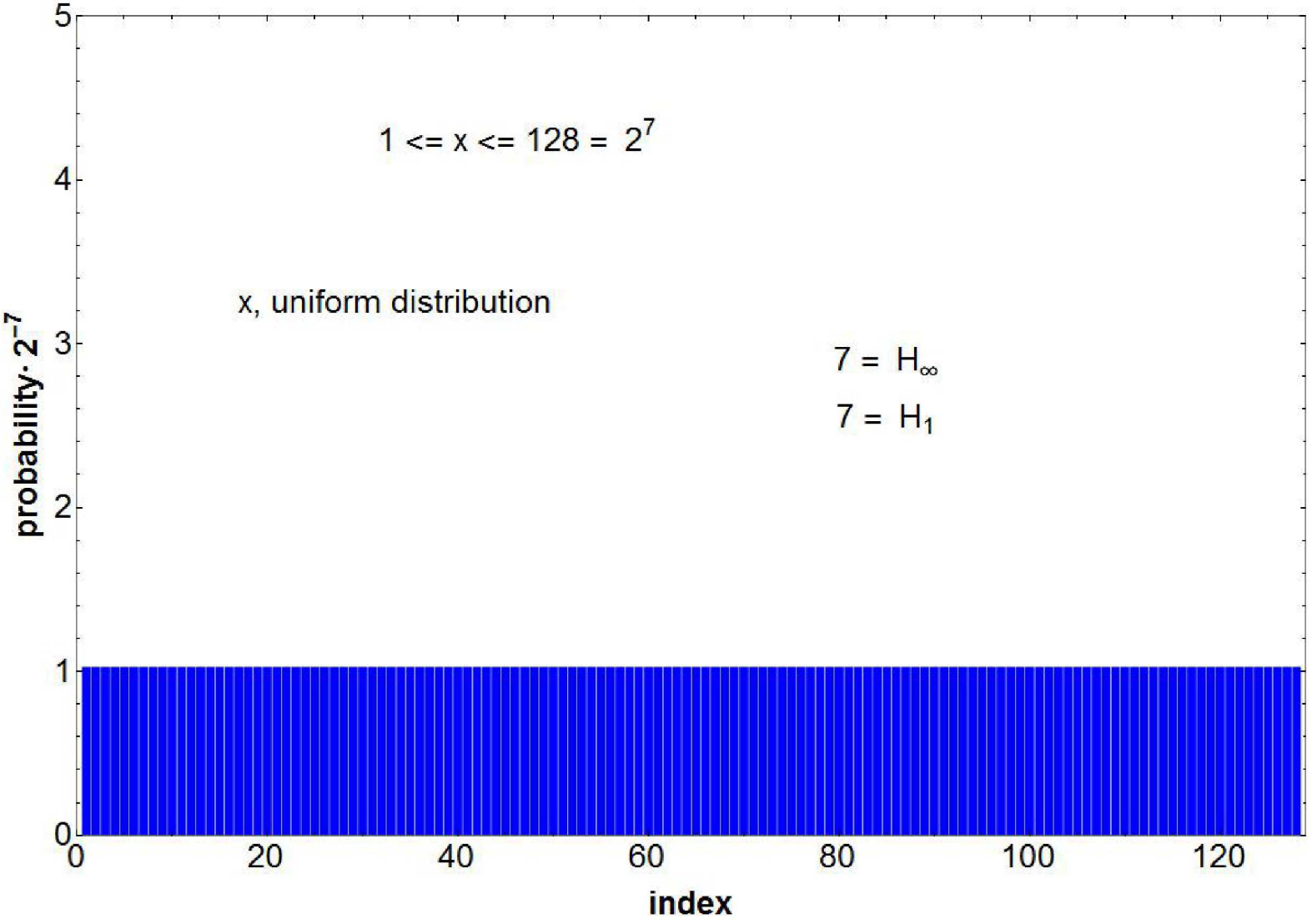


Sandia National Laboratories

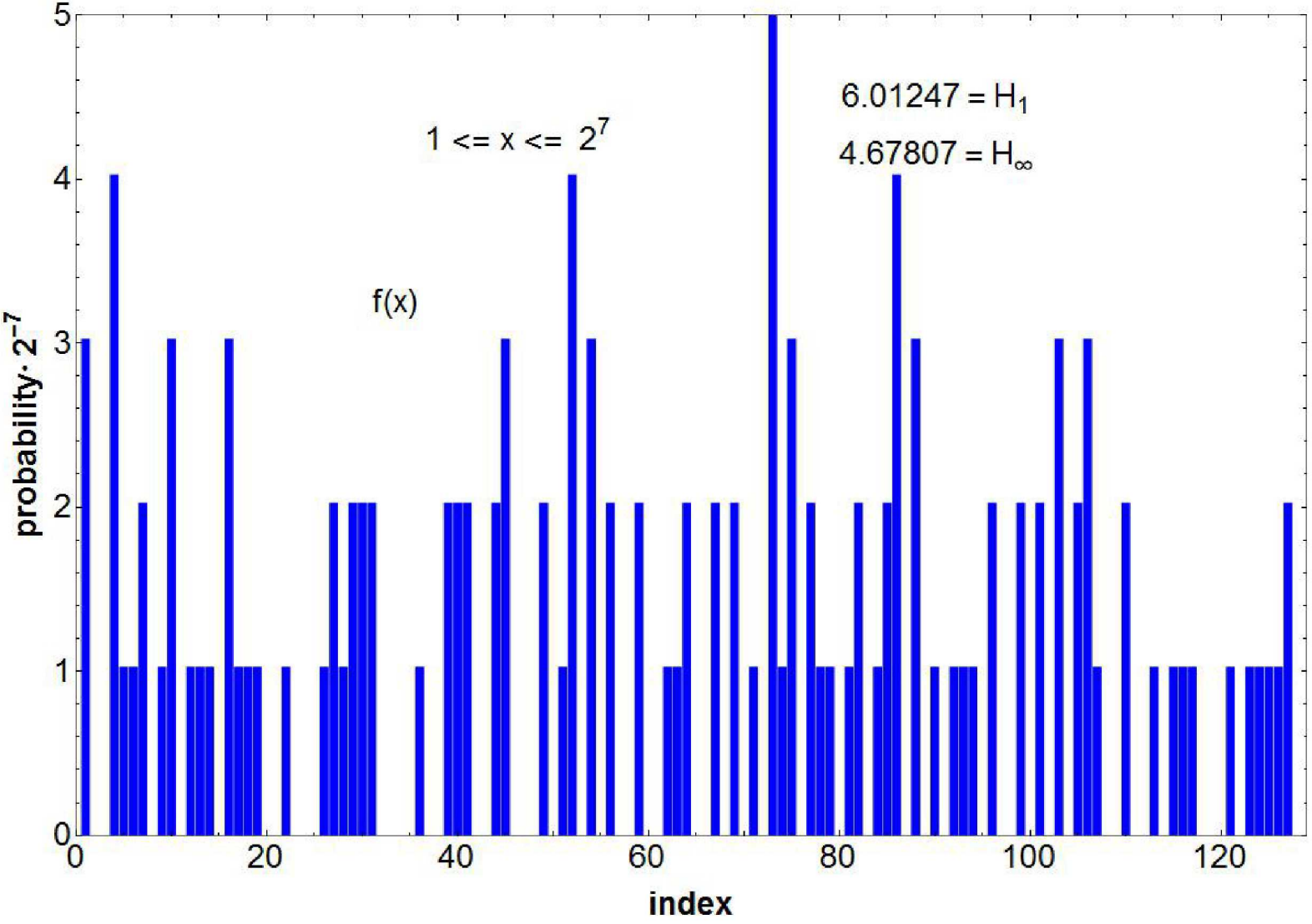
Collisions $\Rightarrow$ Less Entropy



Example - Random Function



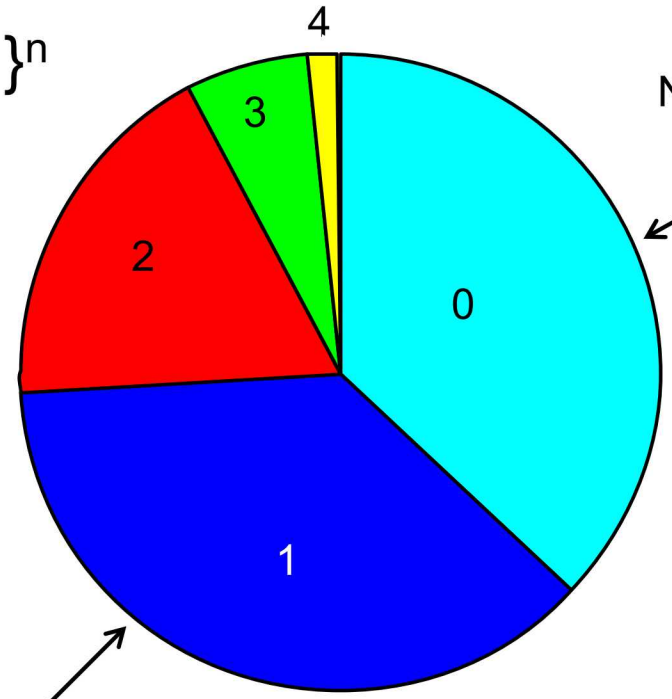
Random Function



Hash Function Collisions

Number of Times Selected

$f: \{0,1\}^n \rightarrow \{0,1\}^n$



No points map to here, $\frac{1}{e}$ of total

$\frac{1}{e} > \frac{1}{3}$

Only one point per value in the range, $\frac{1}{e}$ of total

Hash Function Collisions

$f: \{0,1\}^n \rightarrow \{0,1\}^n$

Assume uniform distribution to start, $H_1 = n$

What is the entropy after hashing once?

n – 0.827245

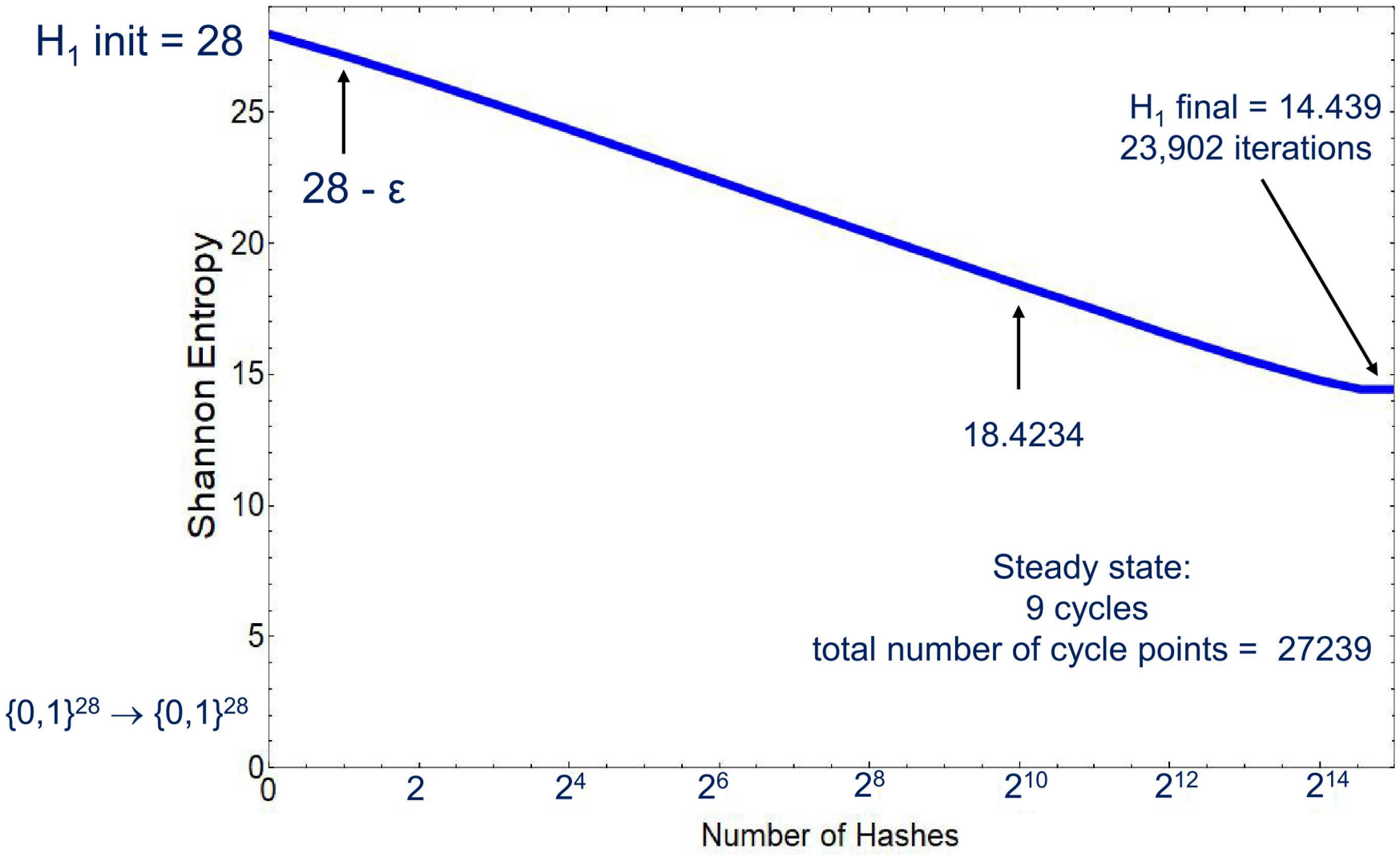
What is the entropy after hashing a million times?

~ n – 20

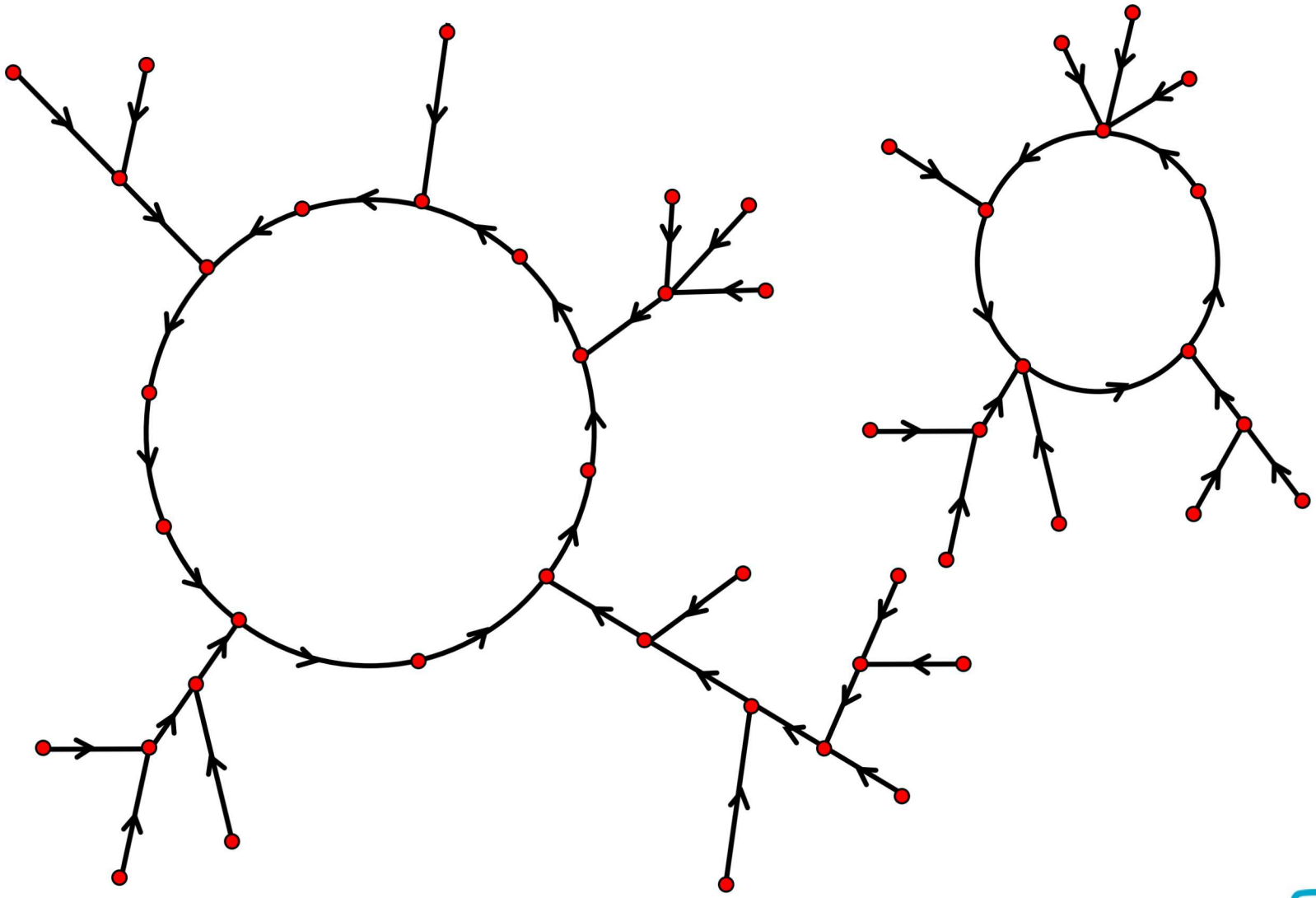
*What is the entropy after hashing **many** times?*

~ n/2

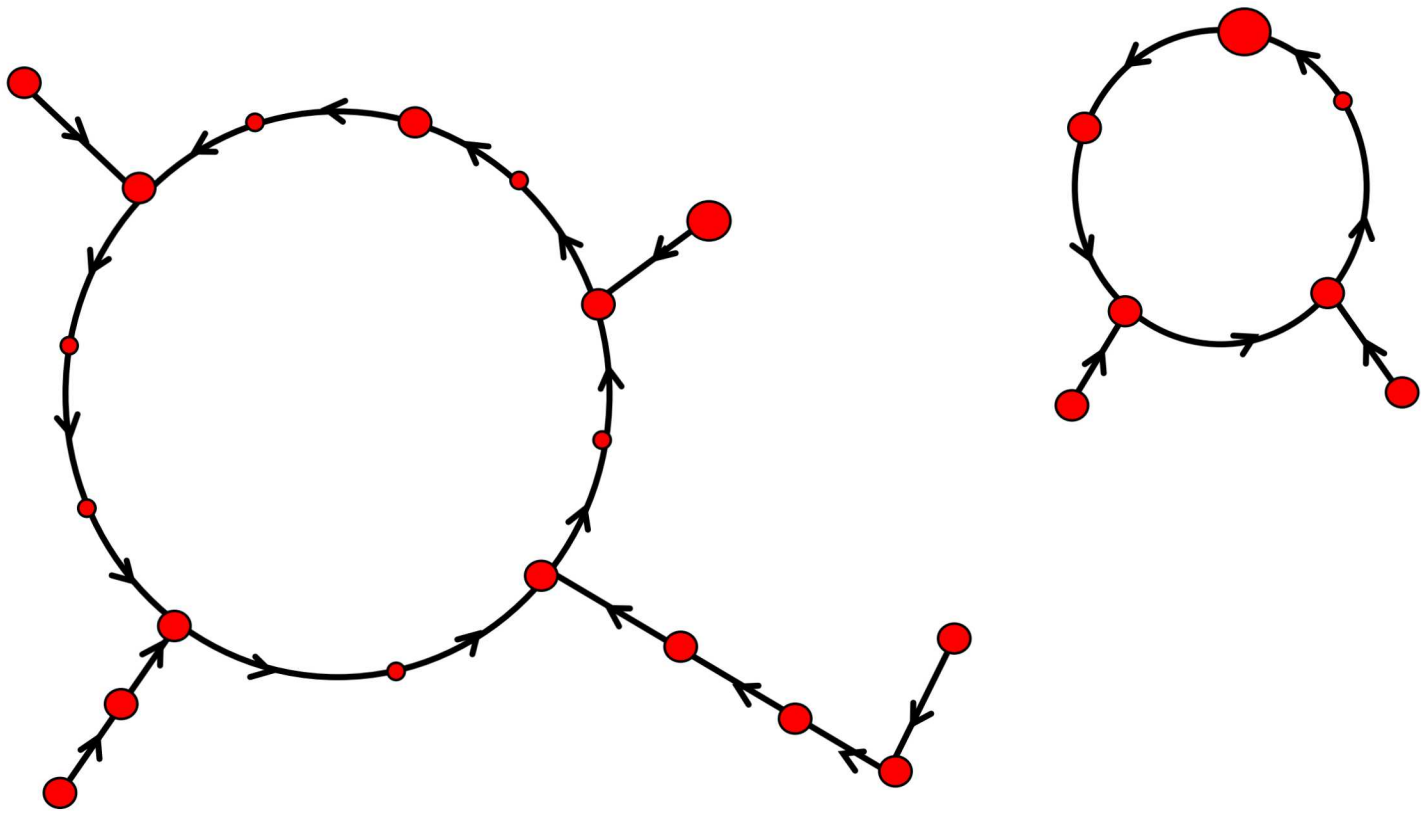
Iterated Hashes



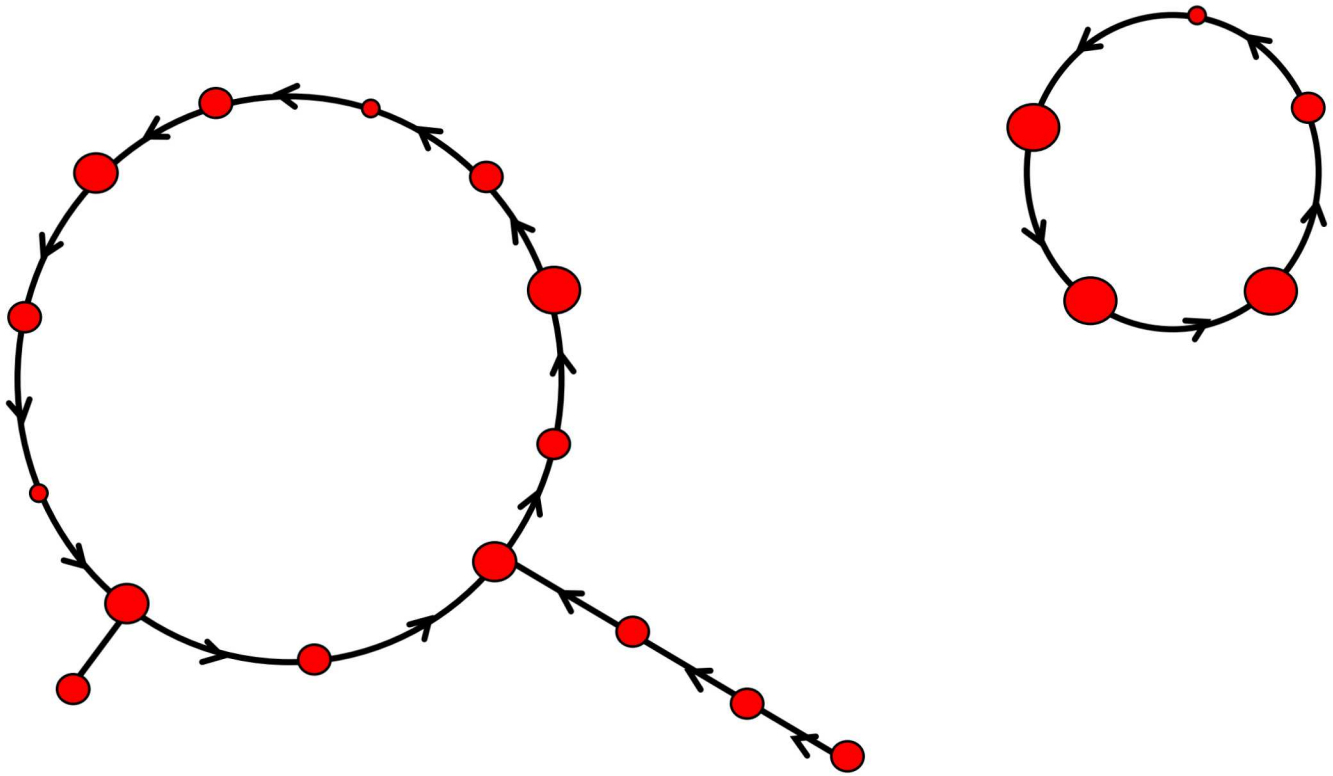
Iterated Hashes



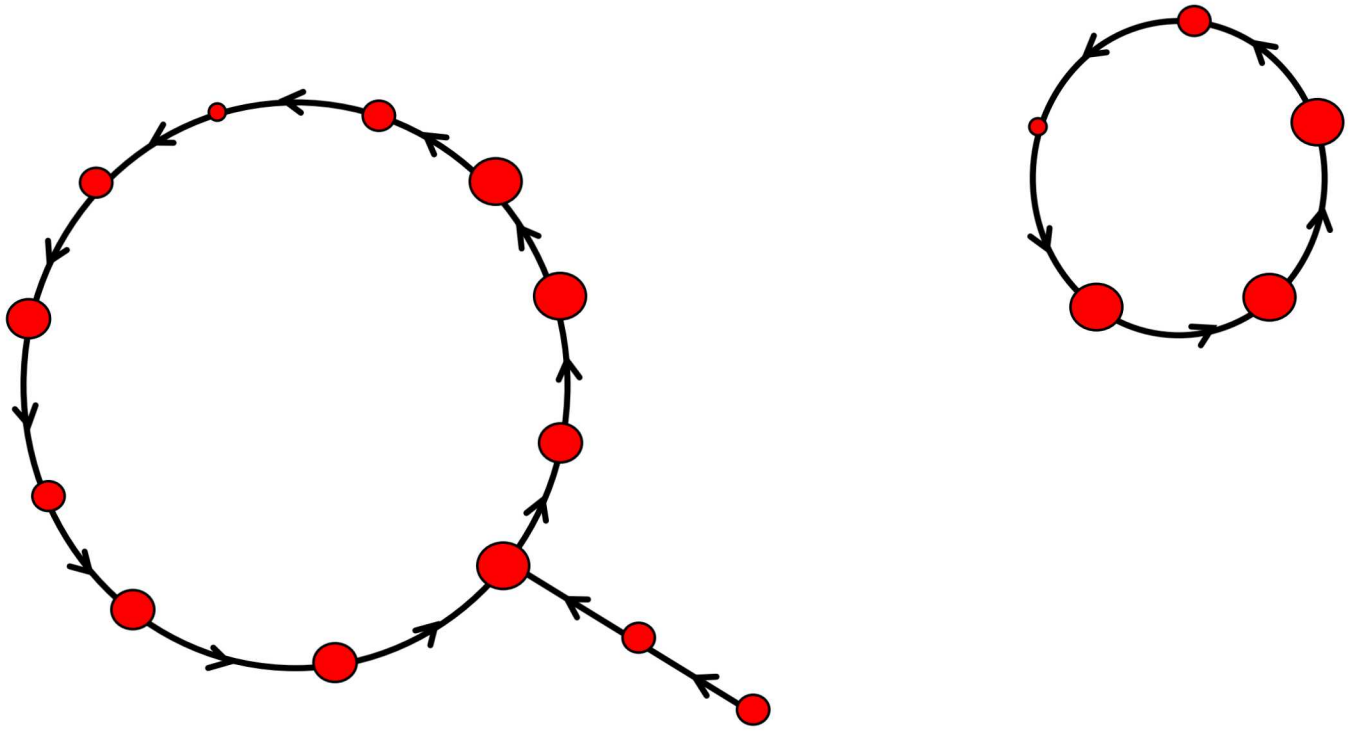
Iterated Hashes



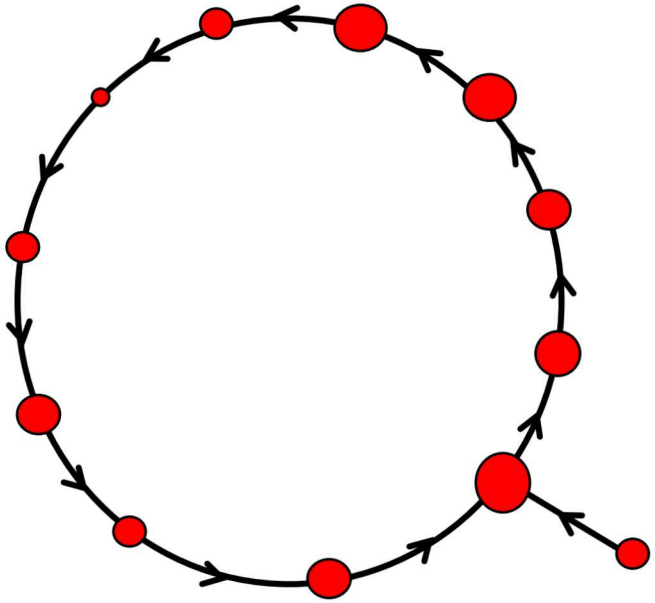
Iterated Hashes

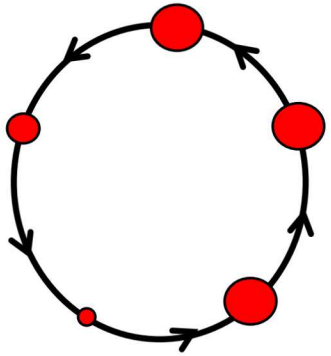


Iterated Hashes

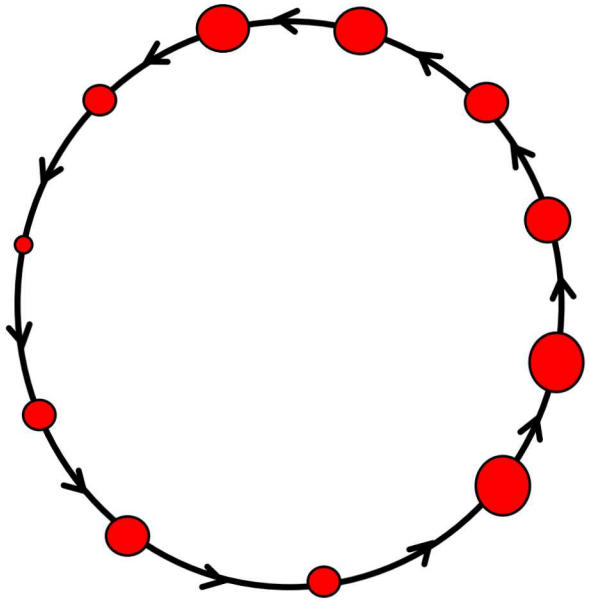


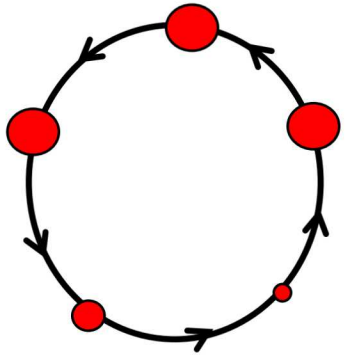
Iterated Hashes





Iterated Hashes





Iterated Hashes

Iterated Hash Entropy Loss

# hashes	H_1	H_∞
1	27.1728	24.5406
2	26.6544	23.7521
2^2	25.9748	22.5737
2^3	25.1674	21.8503
2^4	24.2745	20.8503
2^5	23.3304	19.7379
2^6	22.359	19.2789
2^7	21.3705	18.0871
2^8	20.3812	17.4454
2^9	19.3984	16.6381
2^{10}	18.4221	15.8068
2^{11}	17.4881	15.2744
2^{12}	16.5071	14.7777
2^{13}	15.5954	14.1331
2^{14}	14.7788	13.4554
2^{14+}	14.4390	13.3646

$\{0,1\}^{28}$ -> $\{0,1\}^{28}$

9 cycles
shortest = 3
longest = 12602

Number of cycle points = 27239

Collisions

No function of a distribution can increase its entropy!

Entropy Outline

Motivation and Different Entropy Measures

Collisions & Entropy – Random Functions

Yes/No Questions

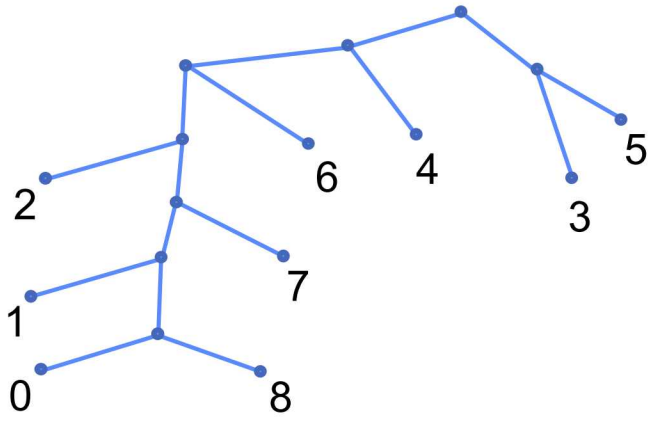
Min-Entropy, Guessing Entropy

Mutual Information

Entropy for keys

PUF Discussion

Fuzzy Extraction



Yes/No Questions

Suppose N = 2^n possible choices, with one correct answer

Q_1 : "Is the answer in the first half of the list?"

Q_2 : "Is the answer in the first half of the reduced list?"

…

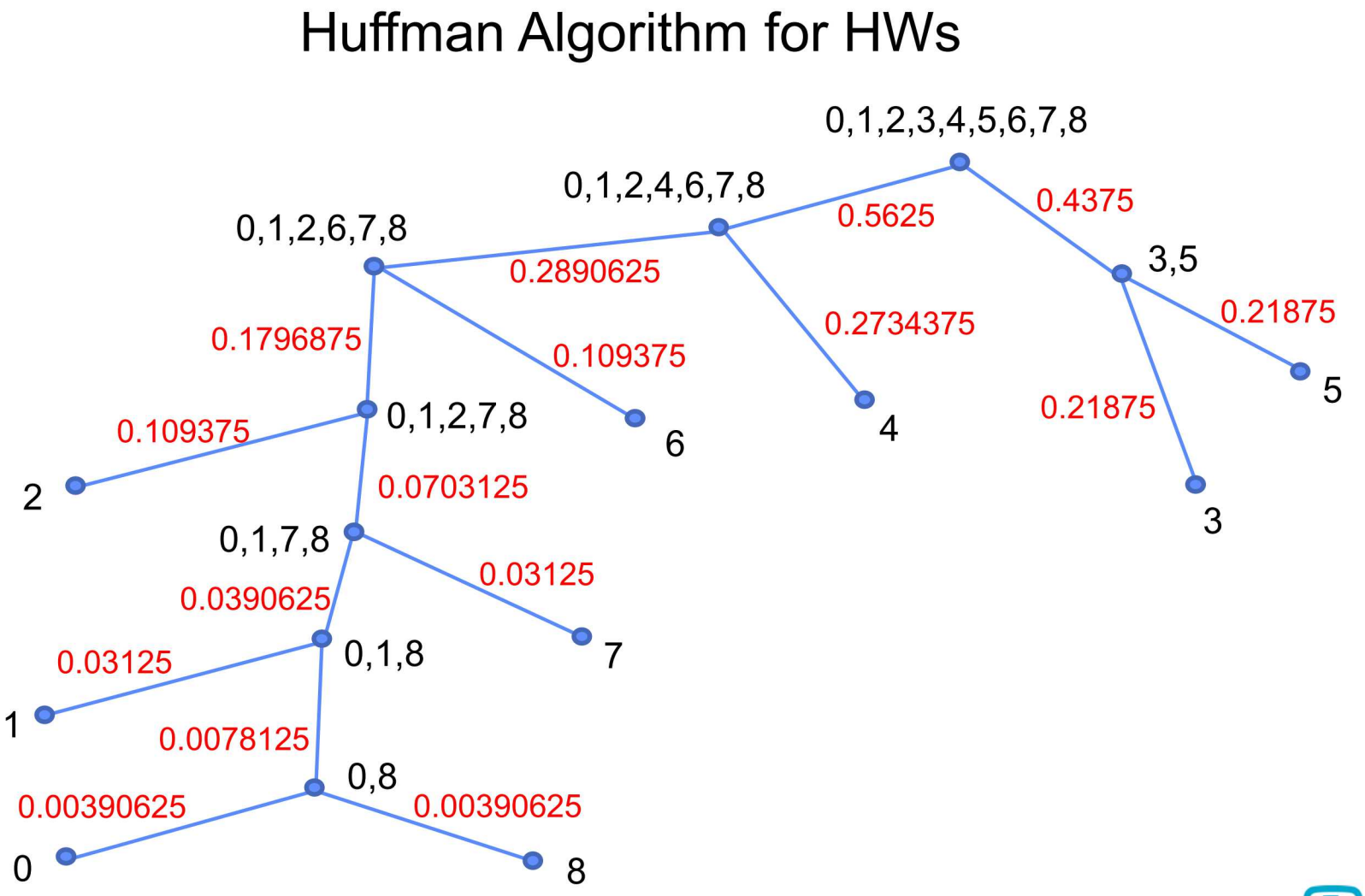
Q_n : "Is the answer the first of the two choices?" *

Number of questions = n ; H_1 = n

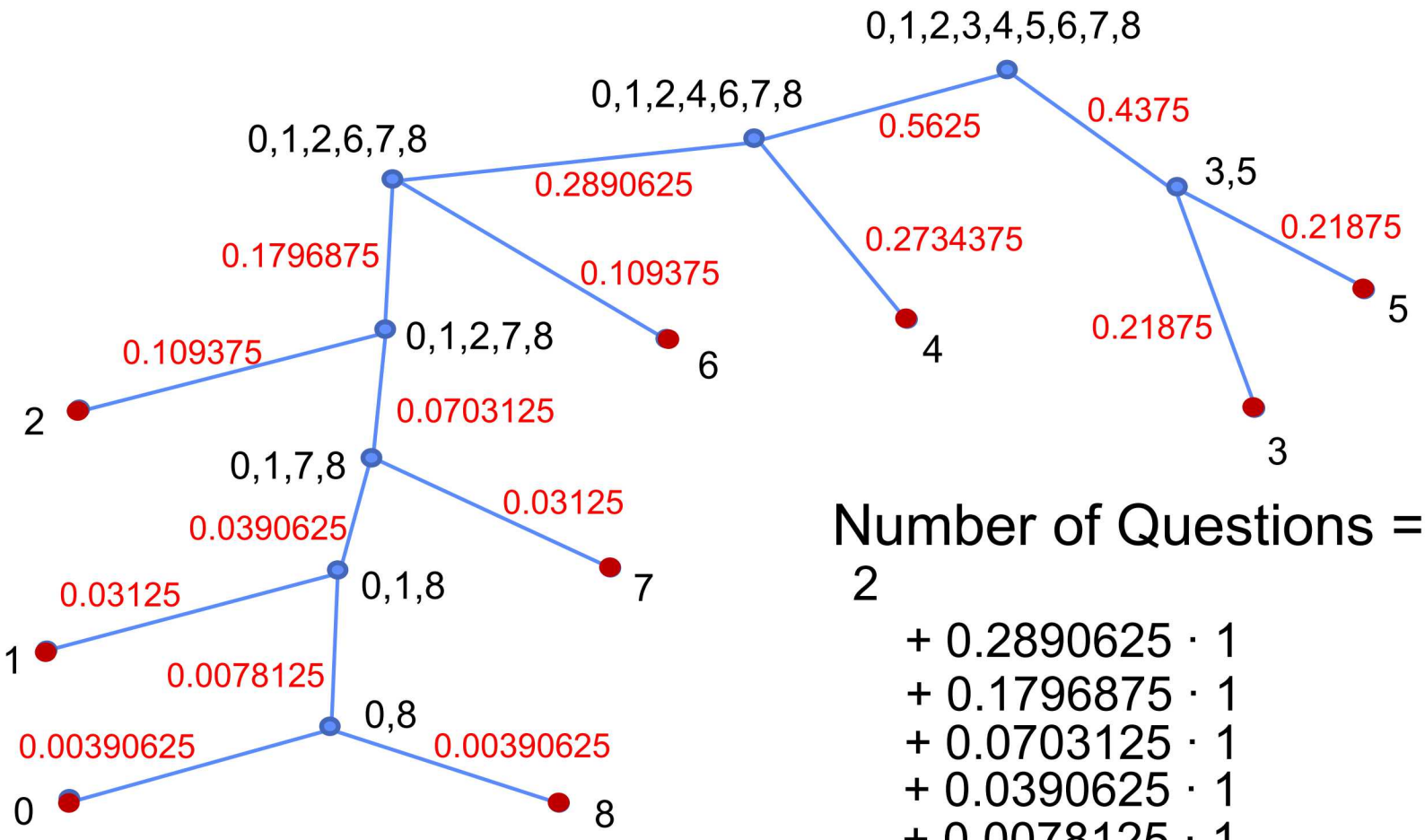
* If the answer was always "No", might want to verify the last possibility

Expected number of *sequential guesses* $\approx \frac{N}{2} = \frac{1}{2} \cdot 2^{H_1}$

Hamming weights



HW Yes/No Questions



Number of Questions =

2

$+ 0.2890625 \cdot 1$
$+ 0.1796875 \cdot 1$
$+ 0.0703125 \cdot 1$
$+ 0.0390625 \cdot 1$
$+ 0.0078125 \cdot 1$

$H_1 = 2.5442$

$= 2.58294$

Entropy Outline

Motivation

Different Entropy Measures

Collisions & Entropy – Random Functions

Yes/No Questions

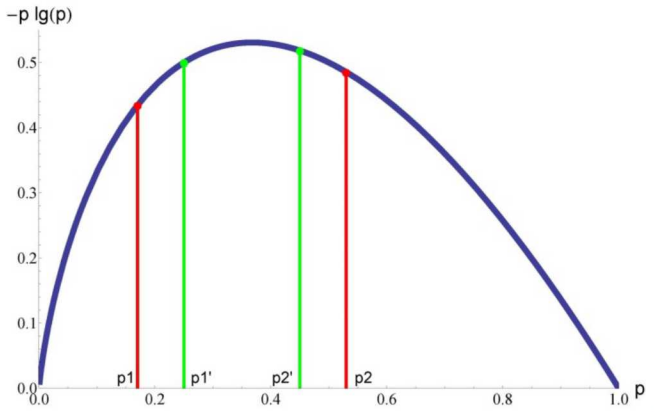
Min-Entropy, Guessing Entropy

Mutual Information

Entropy for keys

PUF Discussion

Fuzzy Extraction



Partial Knowledge

Example

Have 256 bits, but (each bit) only known to probability p

Guess most likely possibilities first

Expected number of guesses to get correct answer $=$

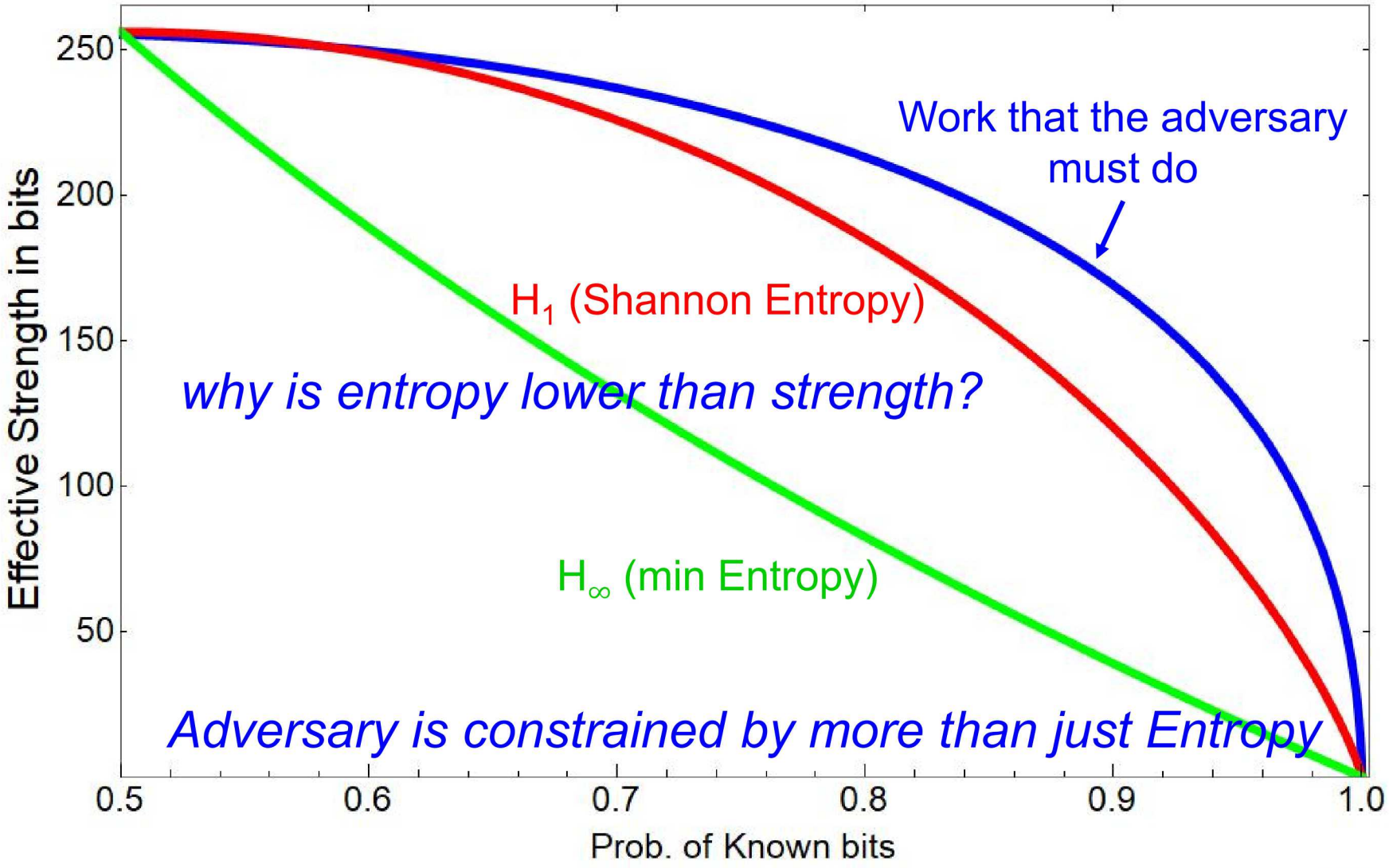
$$E[\text{guesses}] = \sum_{j=1}^{N(k)} p_j \cdot j \qquad \text{(efficient exhaustive search)}$$

probabilities, p_j are decreasing

Bit Probability Uncertainty

	Bit probability	effective strength (bits)
256-bit key	0.50	255.000
	0.55	253.041
	0.60	249.537
	0.65	244.188
	0.70	236.660
	0.75	226.483
	0.80	212.907
	0.85	194.618
	0.90	168.957
	0.95	128.819
	0.99	62.287
	0.999	18.447

Bit Probability Uncertainty



Why Min-Entropy?

Suppose a non-uniform probability distribution, with

2^{425} possible outcomes (keys)

0.40 probability for one outcome (most likely)

2^{425} - 1 remaining possibilities, $p \approx 0.6 / 2^{425}$

$H_1 = 255.9^+$

Expected number of guesses > 2^{423}

"Here is a system with almost 256 bits of entropy; on average, it will take the bad guy more guesses as there are atoms in the universe to get the key!"

"By the way, there's a 40% chance that he'll get the CPI with one guess…"

Why Min-Entropy?

Scenario continued: adversary has 10 parts,

- only needs to defeat one to get the key
- all the keys are different and independent from part-to-part

0.40 probability = most likely key, known to adversary

Strategy : guess most likely key, if wrong, take next part

p = 99.4% that key will be recovered within 10 guesses!

Min-Entropy

Min-Entropy helps to characterize the worst case

Entropy Outline

Motivation and Different Entropy Measures

Collisions & Entropy – Random Functions

Yes/No Questions

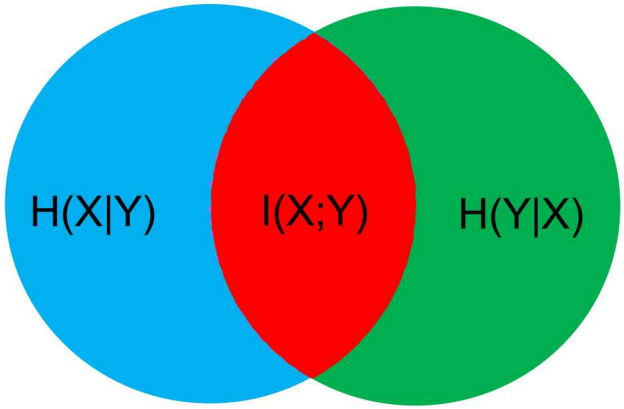
Min-Entropy, Guessing Entropy

Mutual Information

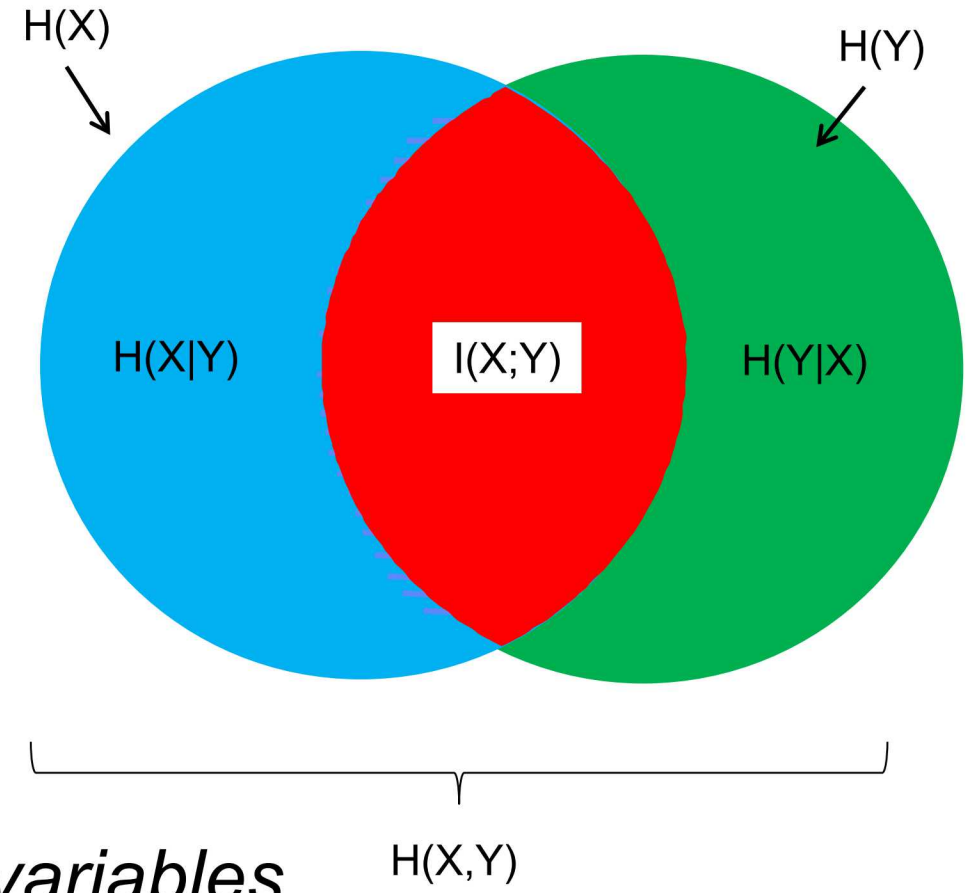
Entropy for keys

PUF Discussion

Fuzzy Extraction



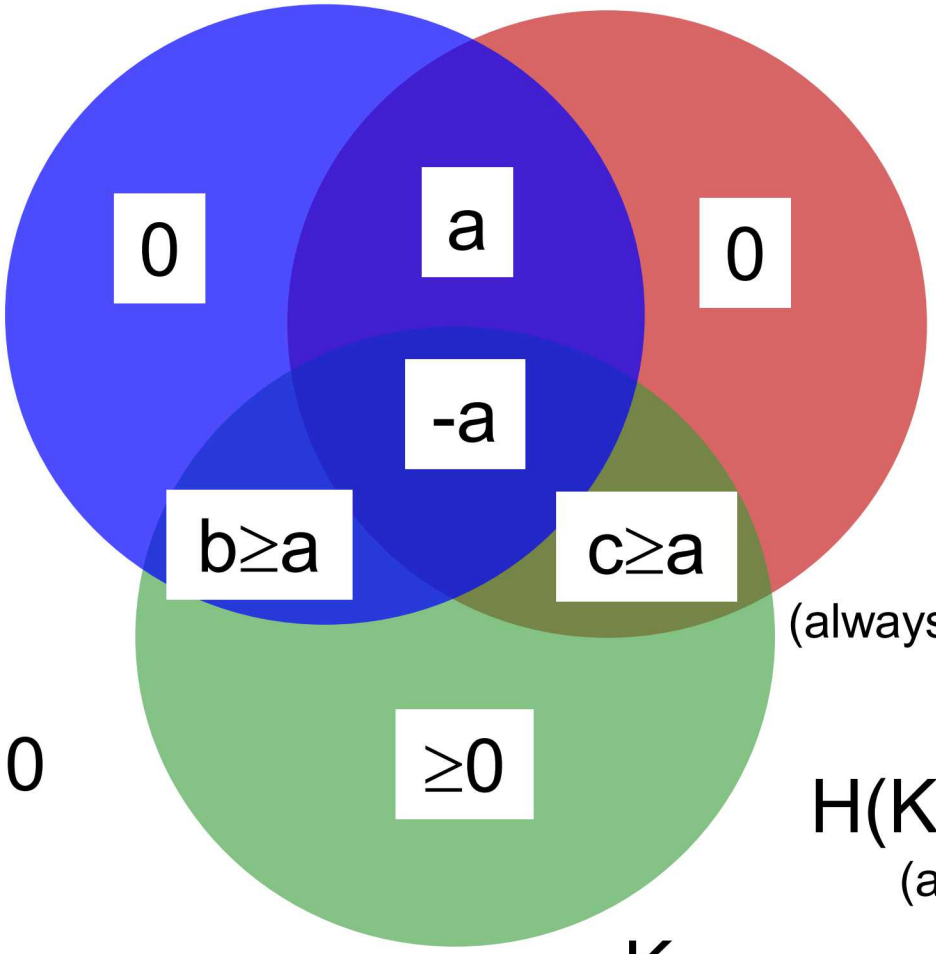
Mutual Information I(X;Y)



$I(X;Y) \geq 0$

For two variables mutual information ≥ 0

Perfect Secrecy Theorem (proof)



I(P;C) = 0
(desired outcome)

H(P | C,K) = 0

I(P;K) = b ≥ 0
I(K;C) = c ≥ 0
(always true for 2 variables)

H(K | P,C) ≥ 0
(always true)

H(K) = (c - a) + b ≥ b = H(P)

AES Results

Using this technique

Assumption:
given a fixed P,
$E_K(P)$ acts as a random function on K (keyspace)

$\Rightarrow$

Given a pt in P,
C has the distribution of the range of a random function

$\varepsilon \approx 0.827$

$$H(K \mid P,C) = \varepsilon$$

less than 1 bit remaining average entropy in key!

Entropy Outline

Motivation and Different Entropy Measures

Collisions & Entropy – Random Functions

Yes/No Questions

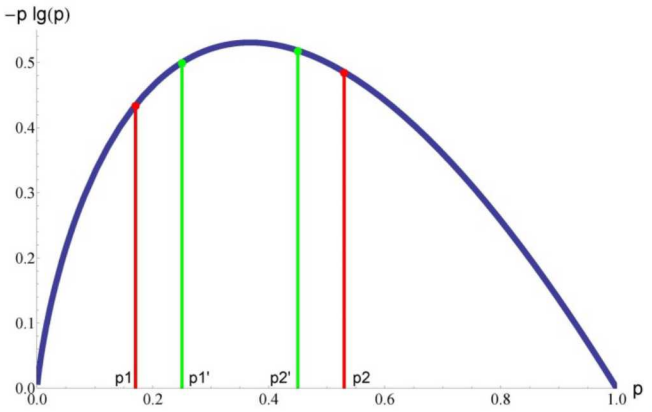
Min-Entropy, Guessing Entropy

Mutual Information

Entropy for keys

PUF Discussion

Fuzzy Extraction

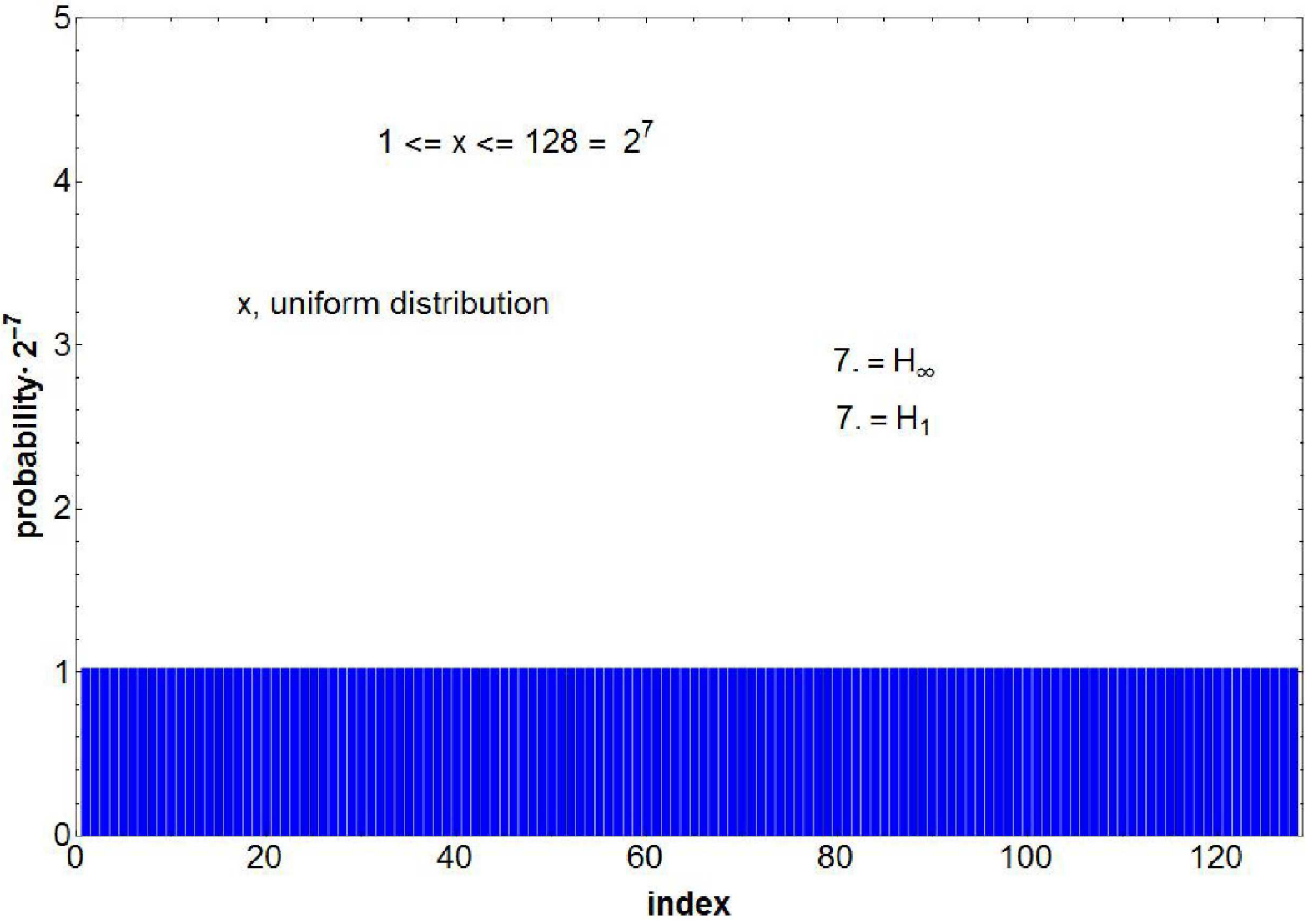


Hash function?

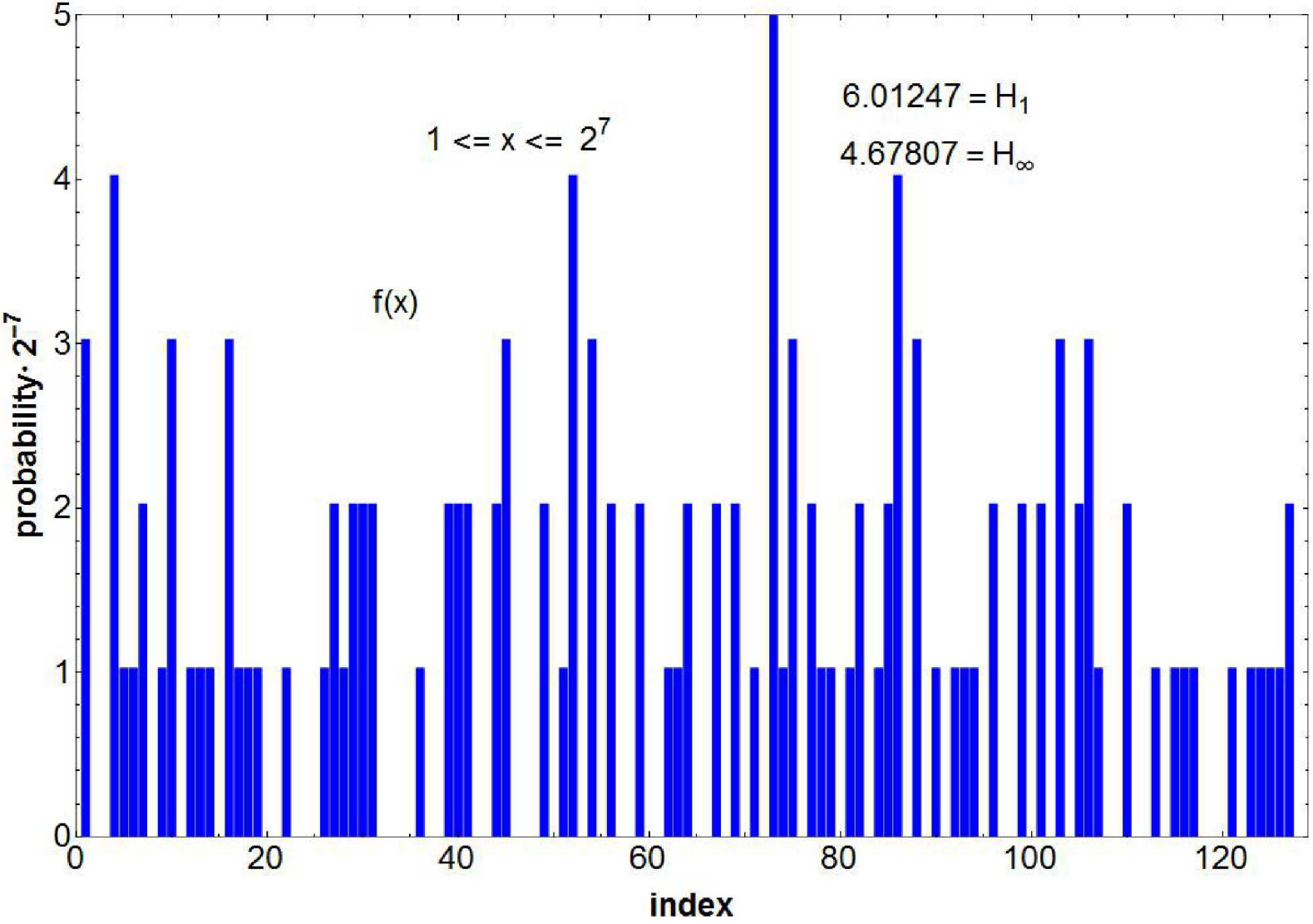
Suppose some initial amount of entropy
Hash the input?

A function of a distribution
***never* increases the entropy**

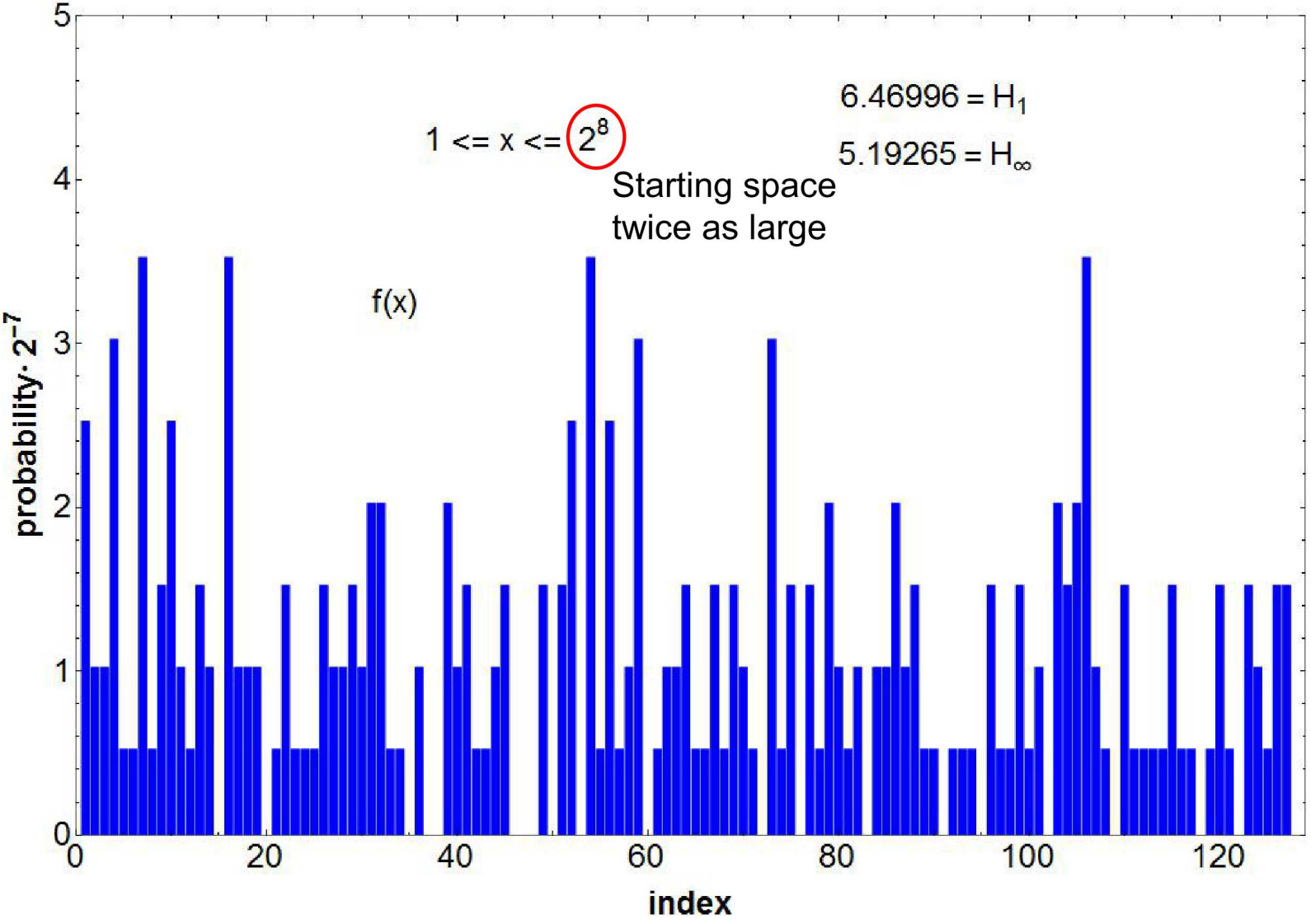
Example - Random Function



Random Function

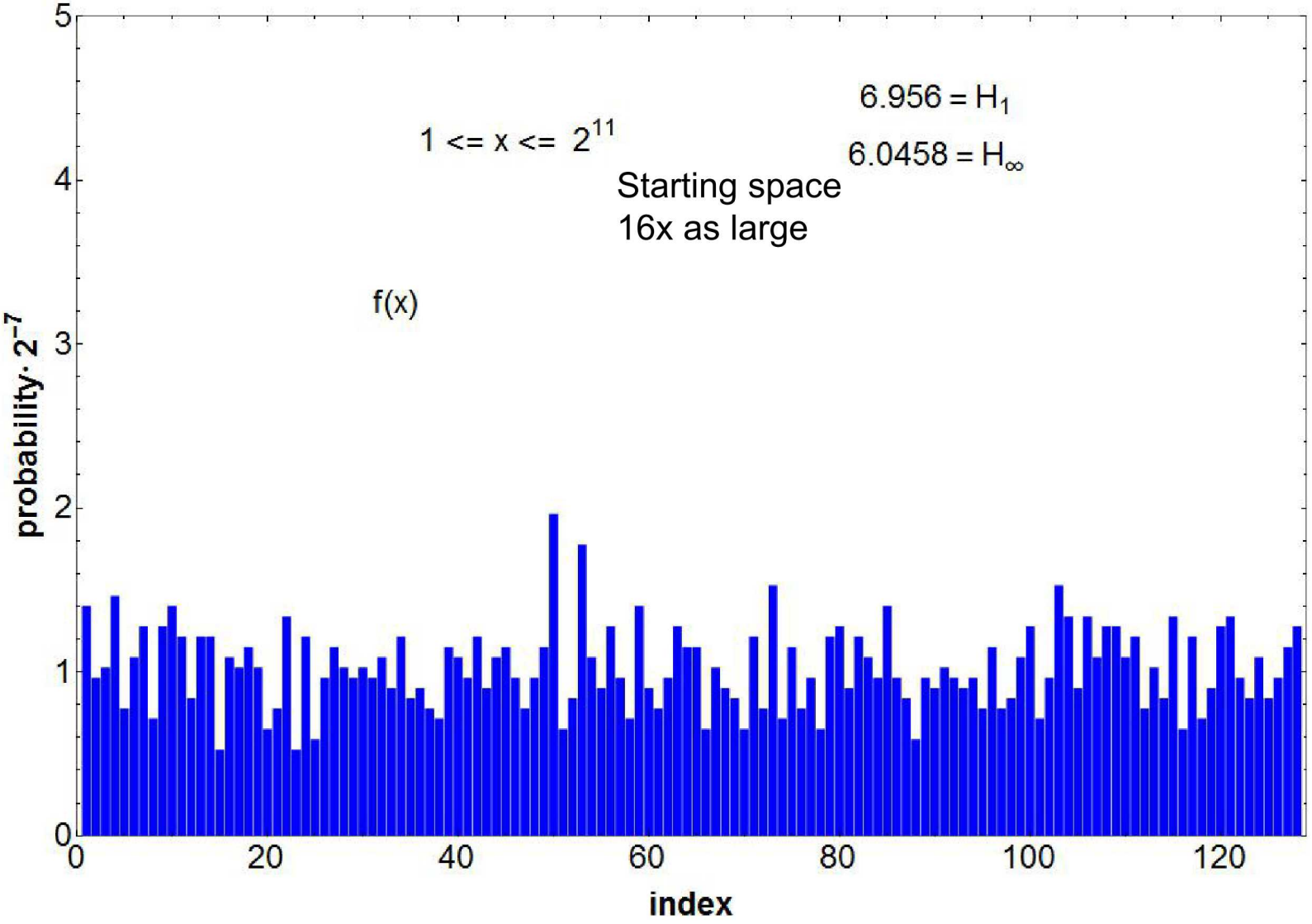


Random Function

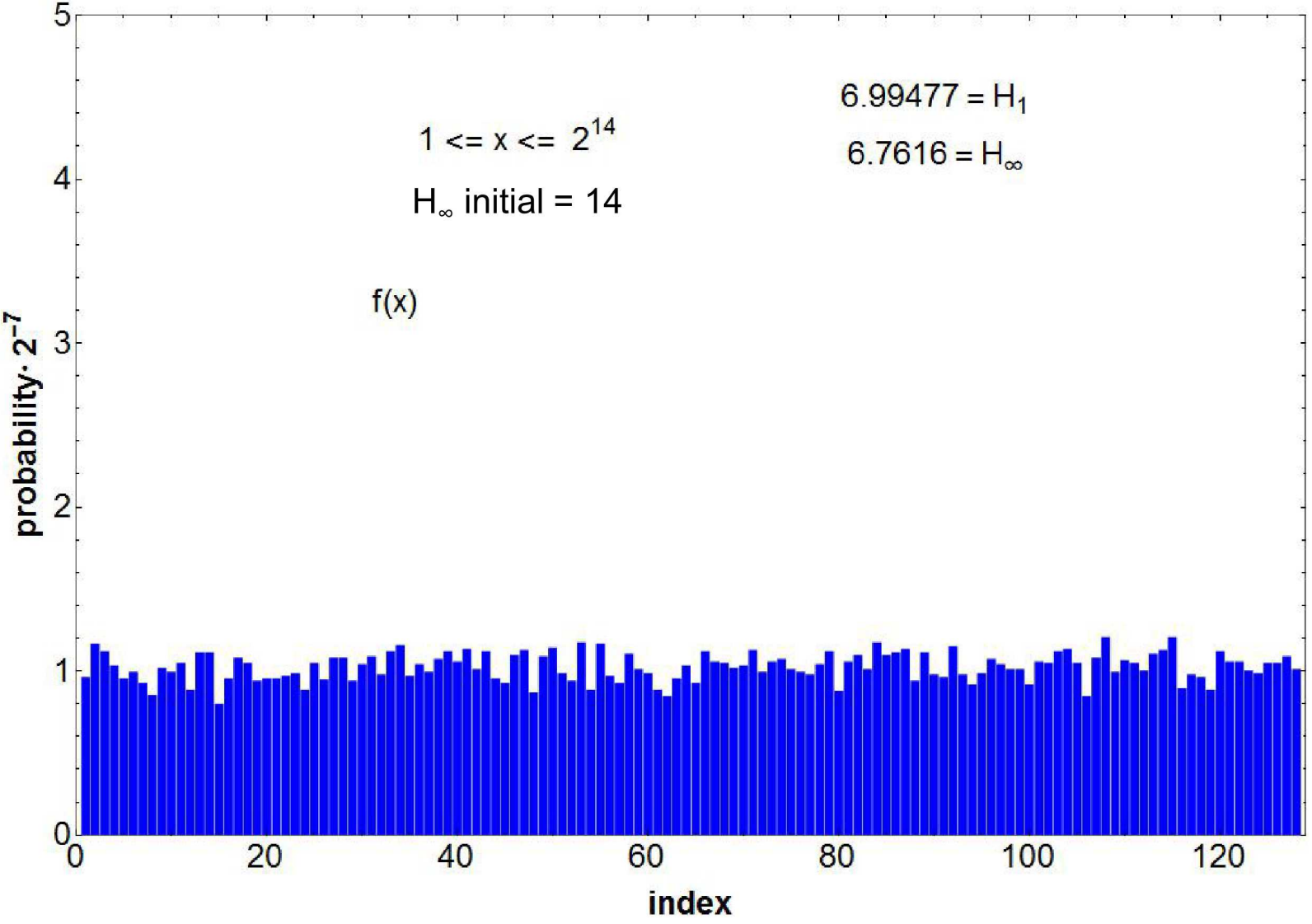


Sandia National Laboratories

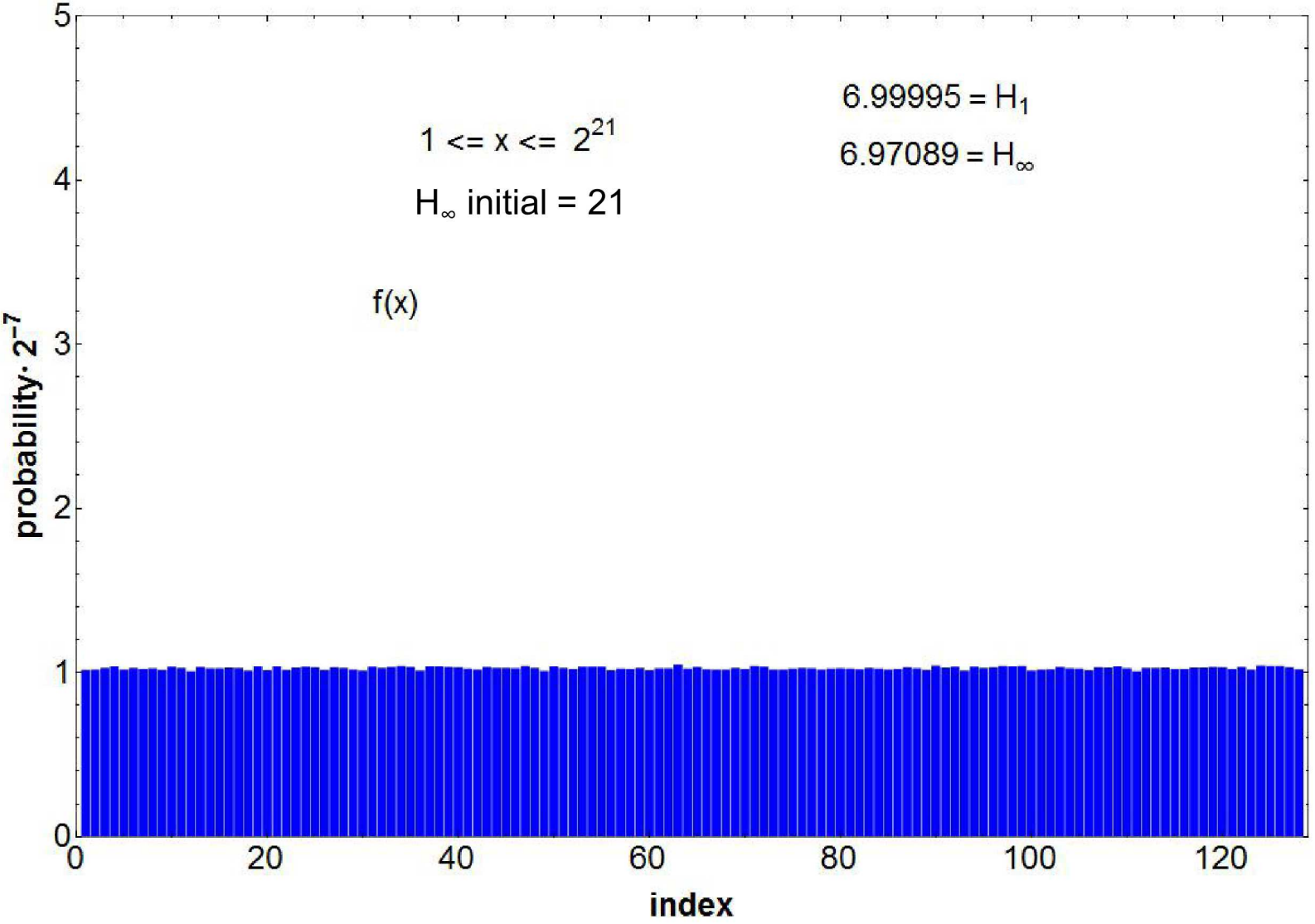
Random Function



Random Function



Random Function



Statistical Distance

Let X and Y be two random variables with range U. Then the statistical distance between X and Y is defined as

$$\Delta(X, Y) \equiv \frac{1}{2} \sum_{u \in U} \left| P[X = u] - P[Y = u] \right|$$

For $\epsilon \geq 0$ we define

$$X \approx_\epsilon Y \Leftrightarrow \Delta(X, Y) \leq \epsilon$$

For example, we might want f(X) $\approx_\epsilon$ Uniform($\{0,1\}^{128}$)

Leftover Hash Lemma

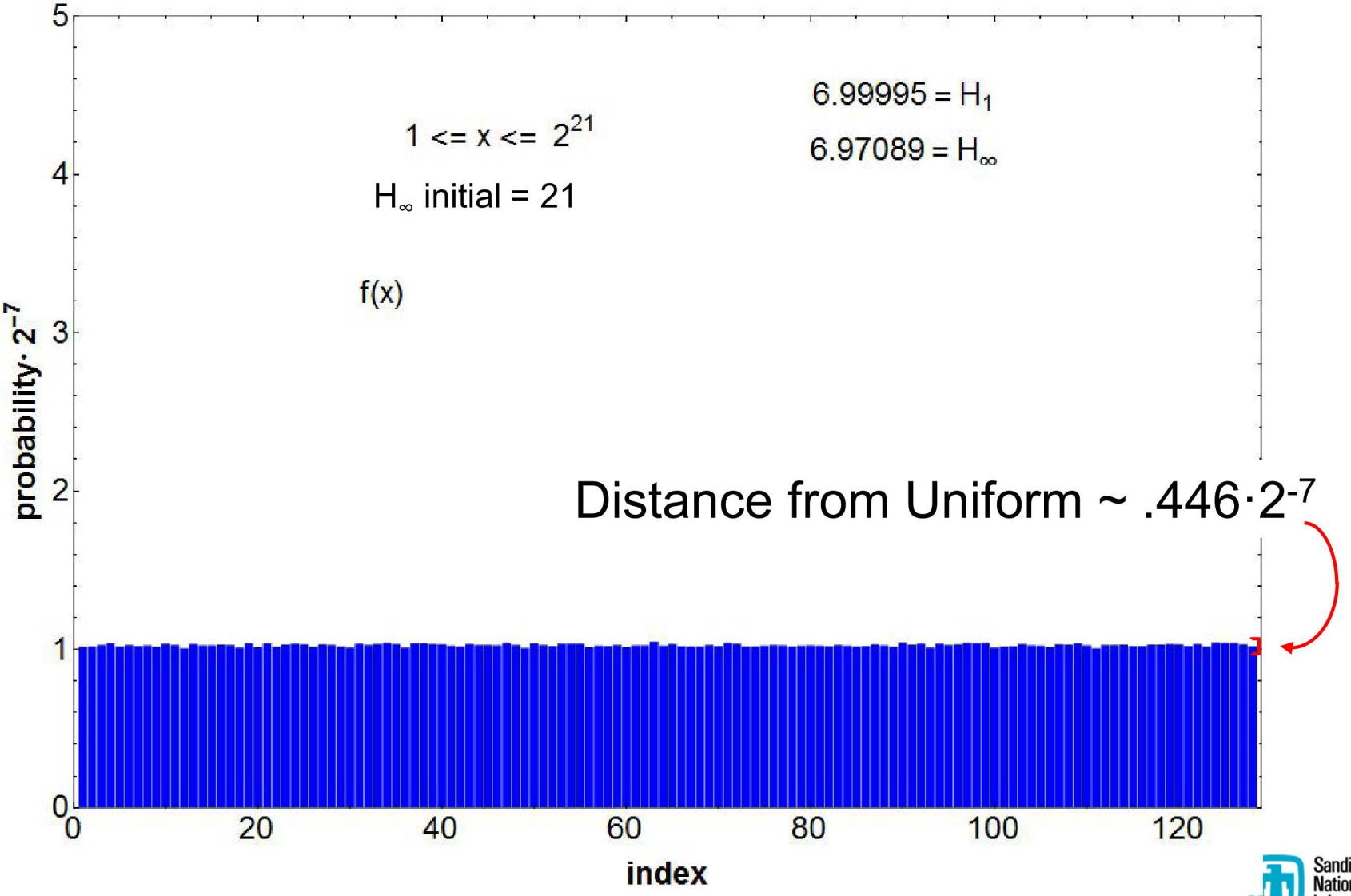
Let $H_\infty(X) \geq k$, Fix $\epsilon > 0$
Let $\mathcal{H}$ be a universal hash family of size 2^d
with output length $m = k - 2 \cdot lg\left(\frac{1}{\epsilon}\right)$

Define $Ext(x, h) = h(x)$

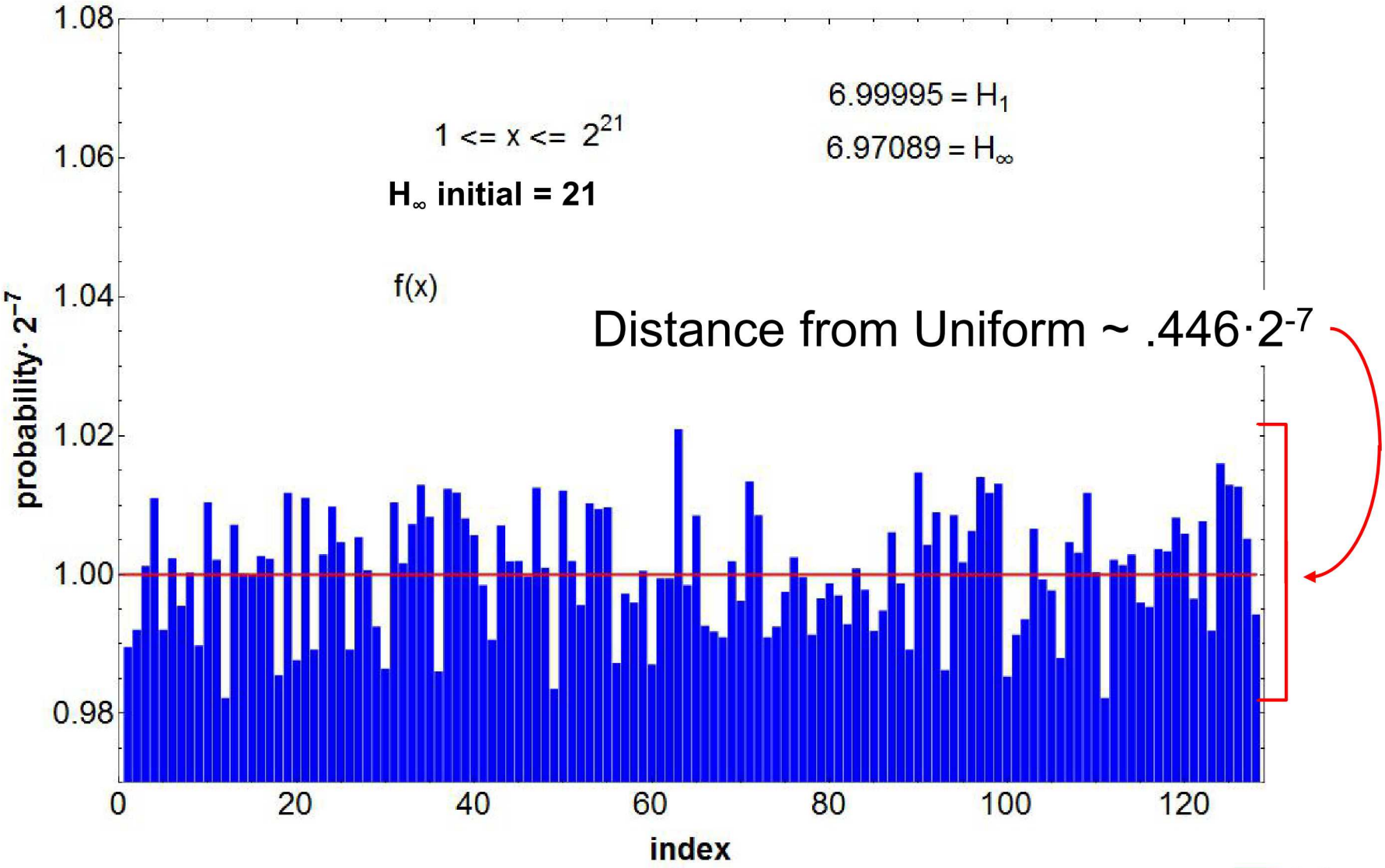
Then *Ext* is a strong $(k, \frac{\epsilon}{2})$ extractor
with seed length d and output length m

m = 256 bits, $\varepsilon = 1/2^{256} \Rightarrow k = 3 \cdot 256 = 768$ bits

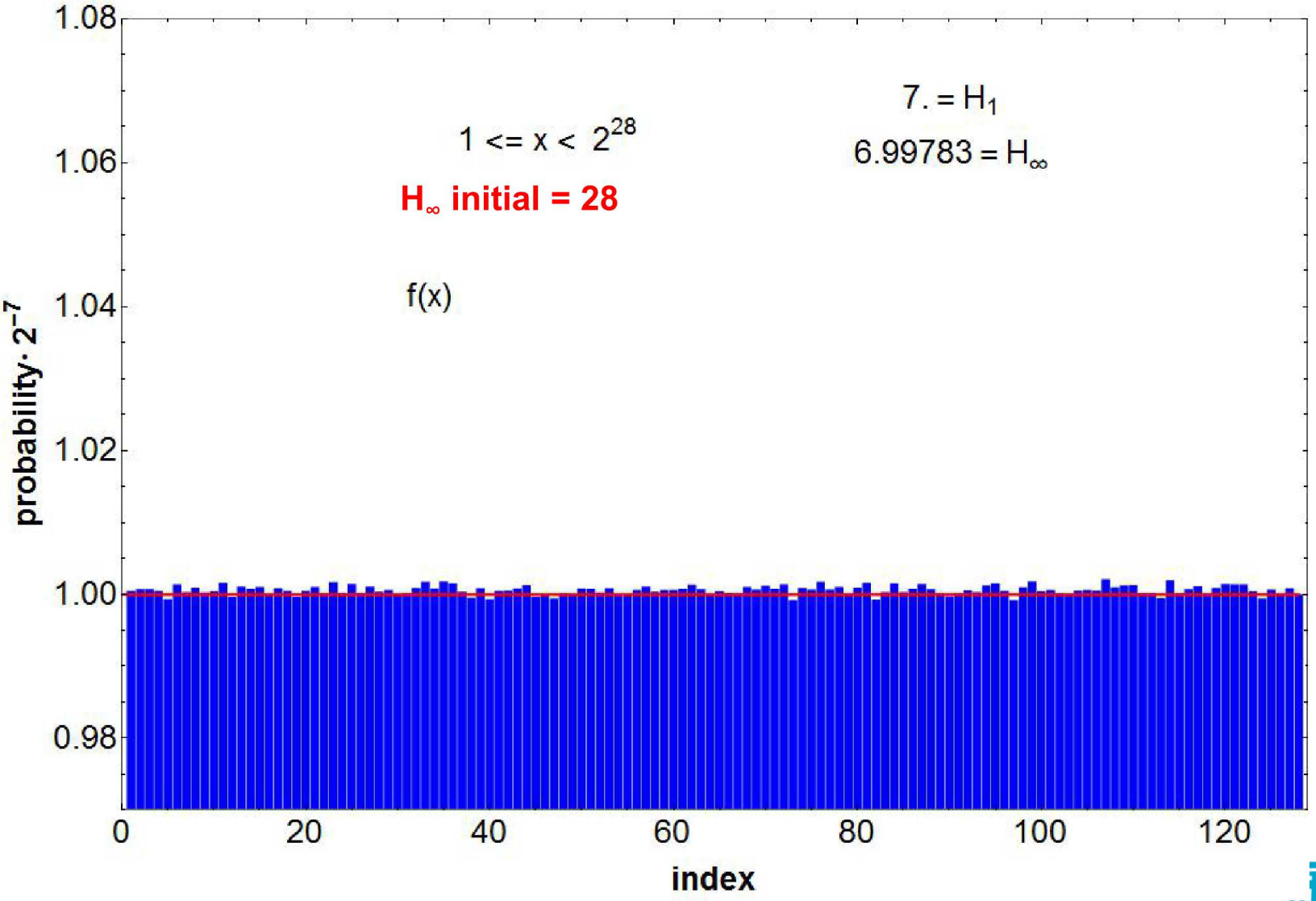
Random Function



(magnified jitter)



(magnified jitter)



Entropy Outline

Motivation and Different Entropy Measures

Collisions & Entropy – Random Functions

Yes/No Questions

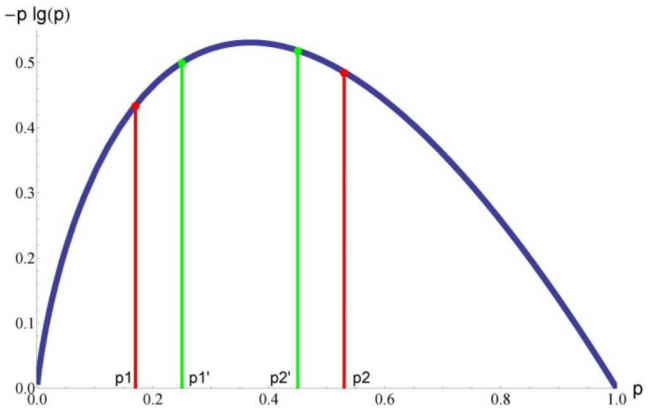
Min-Entropy, Guessing Entropy

Mutual Information

Entropy for keys

PUF Discussion

Fuzzy Extraction



Computing Shannon Entropy

- The Shannon entropy is defined as

$$H_1 = -\sum_{i=0}^{n-1} p_i \cdot lg(p_i)$$

- The p_i's must be known for every event
- The summands must be computed
- The sum must then be evaluated

Computing Shannon Entropy

- Suppose all the information is available

Index	Value	Prob	-lg(Prob)	Product
0	A	.1	3.32	0.332
1	B	.05	4.32	0.216
2	C	.3	1.74	0.522
3	D	.2	2.32	0.464
4	E	.2	2.32	0.464
5	F	.15	2.74	0.411
			H_1	2.409

- Evaluating the sum is straight forward in this case

Computing Shannon Entropy

- Suppose all the information is available

Index	Prob	-lg(Prob)	Product
0	.1	3.32	0.332
1	.05	4.32	0.216
2	.3	1.74	0.522
…	…	…	…
$2^{128}-2$	.2	2.32	0.464
$2^{128}-1$	.15	2.74	0.411
		H_1	???

The moon has on the order of 2^{128} atoms

- Evaluating the sum is NOT straight forward in this case
  - 2^{128} Probabilities must be stored
  - 2^{128} -lg(Prob)'s and Products must be computed
  - 2^{128} things must be added together

This is not computationally feasible

How Many Devices

- Device with three binary PUFs: X_1, X_2, X_3. The 8 outputs may be viewed as:
  - Vectors $(0,0,0), (0,0,1), \ldots, (1,1,1)$
  - Integers $(000), (001), \ldots, (111)$
  - The events do not matter. It is the probability of the events that determine the entropy
- How many devices do you need to confidently determine the probability distribution?
  - Coupon Collector (CC) says $nlg(n)$ to see all values
    - $8 \cdot 3 = 24$
  - CC is not enough to determine the distribution
  - $n^2 = 64?$ Or $n^3 = 512?$
  - Enough to do rudimentary Hypothesis Testing?

How Many Devices are Needed

- Device with 128 binary PUFs: $X_1, \ldots, X_{128}$.
  - Outputs may be viewed as: Vectors or Integers
  - There are up to 2^{128} possible states
  - This many devices does not physically exist
- Device with 32 PUFs: $X_1, \ldots, X_{32}$. Each PUF outputs a four bit value
  - There are also up to 2^{128} possible states
- Device with 8 PUFs: $X_1, \ldots, X_8$. Each PUF outputs a 16 bit value

How do you estimate the probability distribution from insufficient data?

As Hard as Brute Force

- Even if every probability is known, computing the Shannon entropy for a cryptographically relevant situation is as hard brute forcing the key
  - The list of probabilities is the same size as the key space
  - If the probabilities fall into a well known distribution where an entropy formula exists, the formula can be used in place of evaluating the sum
- Acquiring the probabilities for the set of events is as hard as brute forcing the key
  - If these are PUF outputs, you must access more devices than there are keys to determine the probabilities

Assumptions and Simplifications must be made so that estimation can follow

Independence

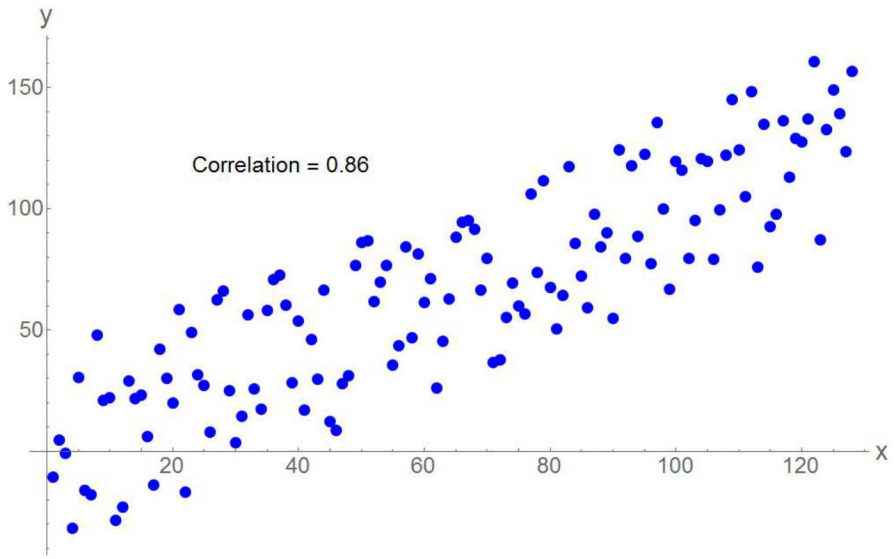
- If $X_1, \ldots, X_{128}$ are independent, then computing H_1 of the joint distribution is straightforward.
  - The entropy of each component is required
  - Data requirements (sample device numbers) are manageable
  - Computational requirements are trivial

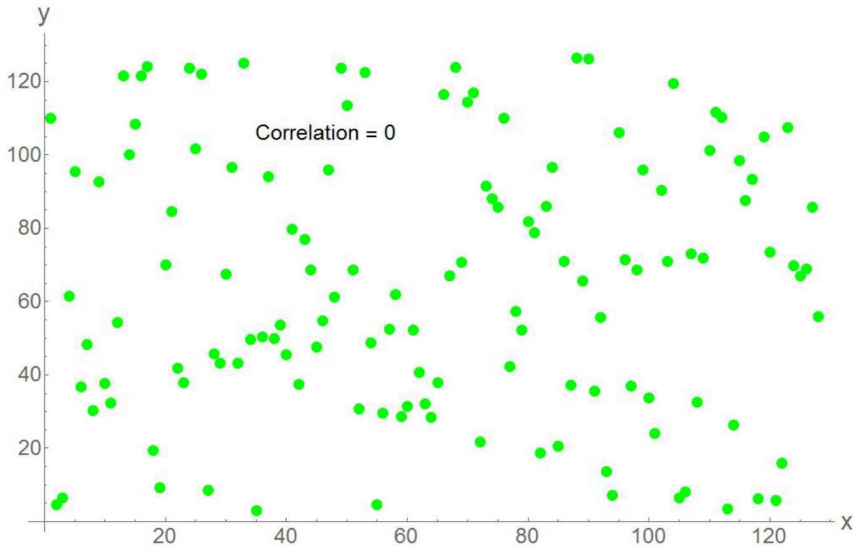
$$H_1(X_1, \ldots, X_{128}) = \sum_{i=1}^{128} H_1(X_i)$$

Independence

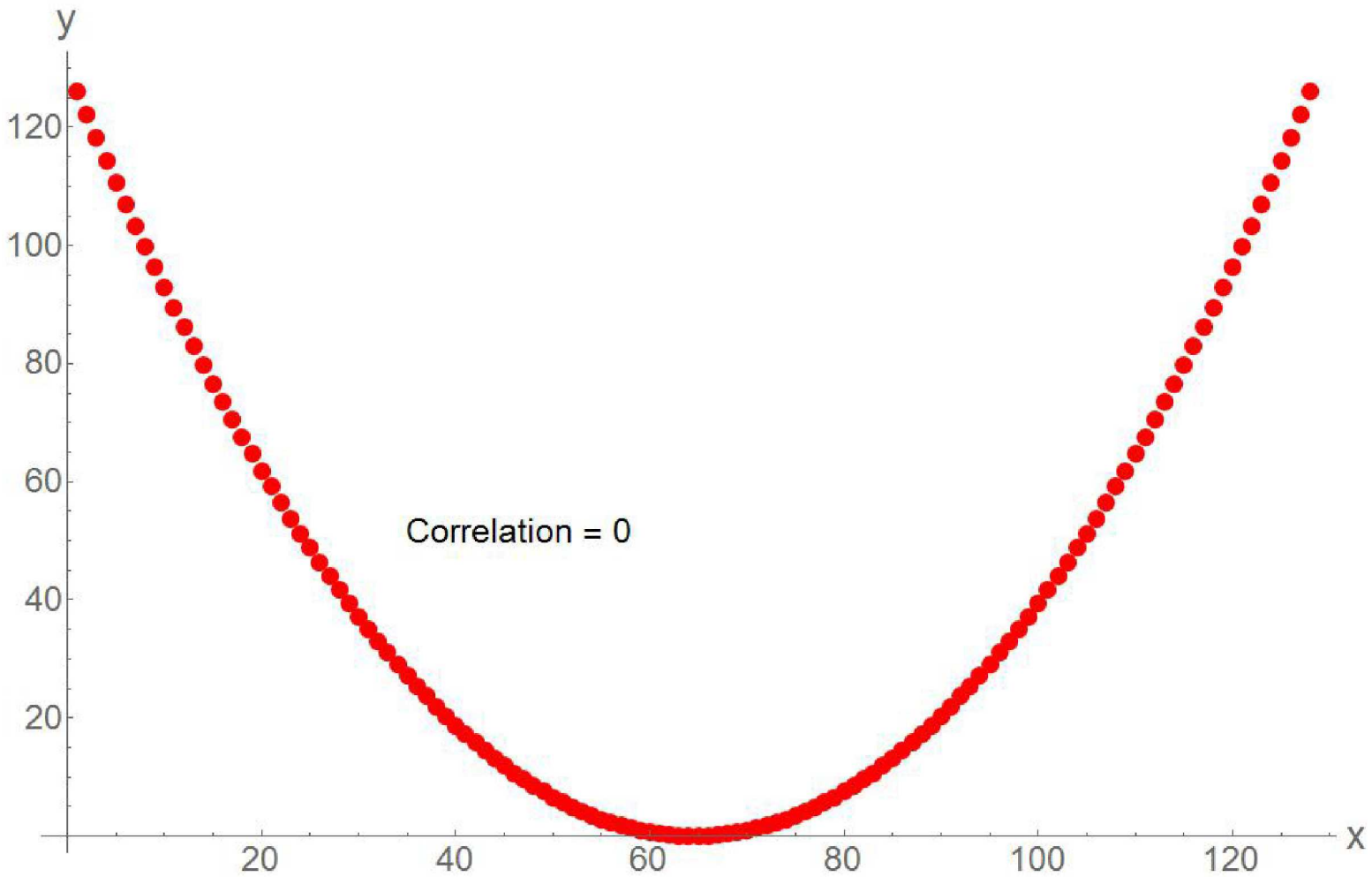
- Independence is a common first assumption
  - It is also almost always false
  - How does one prove or disprove independence of PUFs in a physical system?

Multi-dimensions Correlation





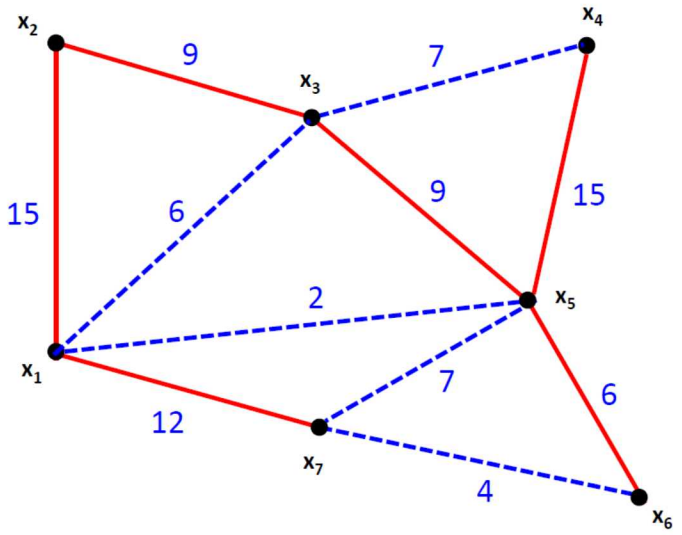
Multi-dimensions Correlation



Large Dimensional Spaces

80 ring oscillators
each has 10 possible values
$p(x_1, x_2, \ldots, x_{80})$ is over $10^{80} \approx 2^{266}$ possible values

How do we calculate H_1?



Active area of research!

Entropy Outline

Motivation and Different Entropy Measures

Collisions & Entropy – Random Functions

Yes/No Questions

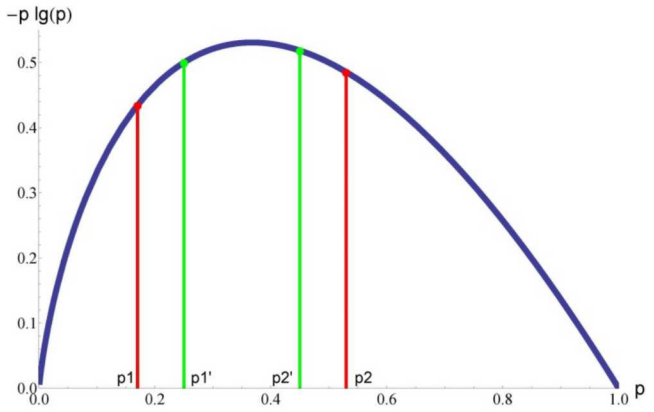
Min-Entropy, Guessing Entropy

Mutual Information

Entropy for keys

PUF Discussion

Fuzzy Extraction



Dodis, Fuzzy Extraction Scheme

- A device uses PUFs to hide a secret seed
- Ingredients
  - Secret seed s that is k bits in length
  - ECC, an (n, k) error correction code
    - n total bits, k data bits
    - $m = n - k$ error correction bits called e
  - A codeword $c = s||e$, which is n bits
  - Secret PUF response p, which is n bits
  - Helper data $w = c \oplus p$, which may be published

The secret s may be a key or a seed for a KDF

Dodis

- During operation
  - Device queries its PUFs to obtain $\hat{p}$
  - Computes $\hat{c} = w \oplus \hat{p}$
  - Error corrects $\hat{c}$ to obtain c $\quad (c = s||e)$
  - Uses c to recover s

- If $p \oplus \hat{p} = \delta$ has small Hamming weight
  - $\hat{c} = w \oplus \hat{p} = w \oplus p \oplus \delta = c \oplus \delta$
  - $\hat{c} = c \oplus \delta$ is correctible by the ECC to recover c

Reveal the Helper Data

- Let
  - P be the set of all n-bit PUF values
  - S be the set of all k bit seeds
  - C be the set of all codewords
  - W be the set of all possible n-bit helper data
- Assume that $H_1(S) = k$, that is, the seeds are drawn uniformly from S
- If the helper data is public, what is the remaining entropy in the system?

What is $H_1(C, P|W)$?

What is $H_1(C, P|W)$?

W is fully dependent on (C, P)

- $H_1(C, P, W) = H_1(C, P)$

C and P are independent

- $H_1(C, P) = H_1(C) + H_1(P)$

We have $H_1(C) = H_1(S)$

So

$$\begin{aligned} H_1(C, P|W) &= H_1(C, P, W) - H_1(W) \\ &= H_1(S) - [H_1(W) - H_1(P)] \\ &= k - [H_1(W) - H_1(P)] \\ &\leq k \end{aligned}$$

It can be shown that $H_1(W) \geq H_1(P)$

A Few Cases

Suppose $H_1(P) = n$

- $n \geq H_1(W) \geq H_1(P) \geq n$
- So $H_1(W) - H_1(P) = 0$
- $H_1(C, P|W) = k$ which is the entropy in the seed space

If the PUFs have full entropy, no information about the seed is given away by revealing the helper data

If $H_1(P) < n$, then it may be that $\delta = H_1(W) - H_1(P) > 0$

- $H_1(C, P|W) = k - \delta < k$

The value of δ is tied strongly to the interplay between the PUFs and the error correction code

Dodis Lower Bound

Suppose $H_1(P) = n - \mu$ We have that

- $H_1(C, P|W) = k - [H_1(W) - n + \mu] \geq k - \mu$

This is a lower bound on the system's entropy

Details matter, however…

For a wide variety of PUF distributions and Error Correction Codes the assumption that $H_1(W) \approx n$ is reasonably close. If so and $k \geq \mu$, we have

$$H_1(C, P|W) = k - \mu$$

Any deficiency of entropy in P translates to a reduction of entropy in system.

References

Cover & Thomas, *Elements of Information Theory, 2nd ed.*,
J. Wiley & Sons; Hoboken, NJ; 2006.
ISBN 978-0-471-24195-9

Robert Gray, *Entropy and Information Theory, 2nd ed.*,
Springer-Verlag; New York, NY; 2011.
ISBN 978-1441979698

Katz & Lindell, *Introduction to Modern Cryptography, 2nd ed.*,
Chapman & Hall/CRC; Boca Raton, FL; 2015.
ISBN 978-1-4665-7026-9