A Challenge Code for Maximizing the Entropy of PUF Responses

Institut
Mines-Télécom

Olivier Rioul[1], Patrick Solé[1],
Sylvain Guilley[1,2] and Jean-Luc Danger[1,2]

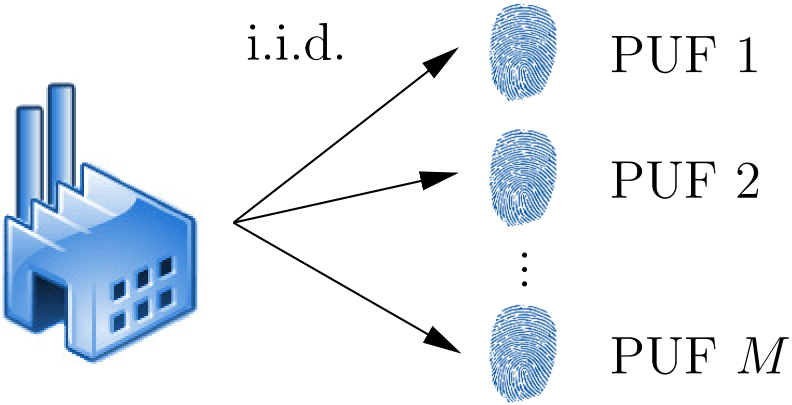
[1] LTCI, CNRS, Télécom ParisTech,
Université Paris-Saclay, 75 013 Paris, France.
Email: firstname.lastname@telecom-paristech.fr

[2] Secure-IC S.A.S., 15 Rue Claude Chappe, Bât. B,
ZAC des Champs Blancs, 35510 Cesson-Sévigné, France.
Email: firstname.lastname@secure-ic.com

<sylvain.guilley@telecom-paristech.fr>

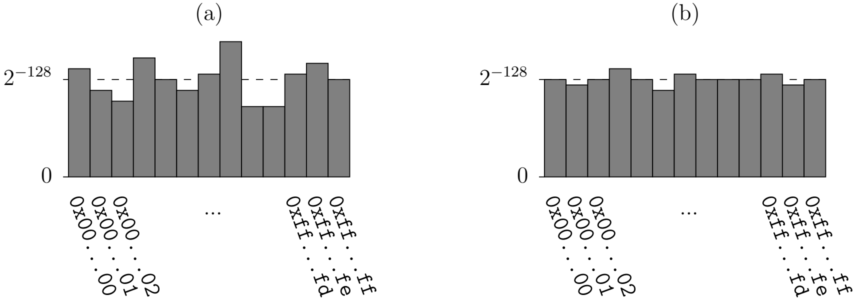
Outline

Entropy of PUFs



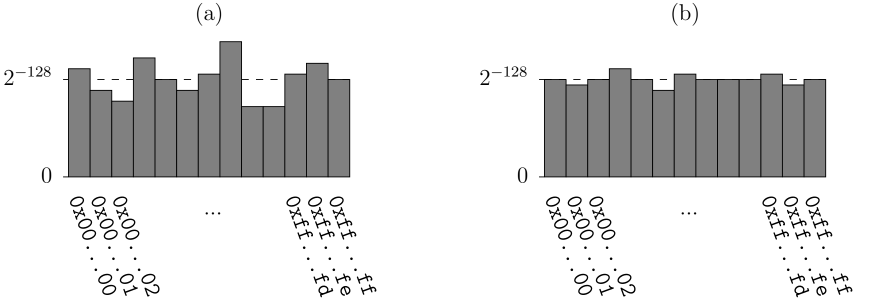
PUFs are instanciations of blueprints by a fab plant

After fabrication (estimation $\hat{\mathbb{P}}$)



Which PUF is the most entropic?

After fabrication (estimation $\hat{\mathbb{P}}$)



Which PUF is the most entropic?

$$\text{Recall } H = \sum_{c=\texttt{0x00...00}}^{\texttt{0xff...ff}} \mathbb{P}(R = \mathsf{PUF}(c)) \log \mathbb{P}(R = \mathsf{PUF}(c)).$$

Before fabrication

- **Stochastic model**
- Active discussion at ISO sub-committee 27:

ISO/IEC JTC 1/SC 27/WG 3 **N1233**

REPLACES:

ISO/IEC JTC 1/SC 27/WG 3

Information technology - Security techniques - Security evaluation, testing and specification

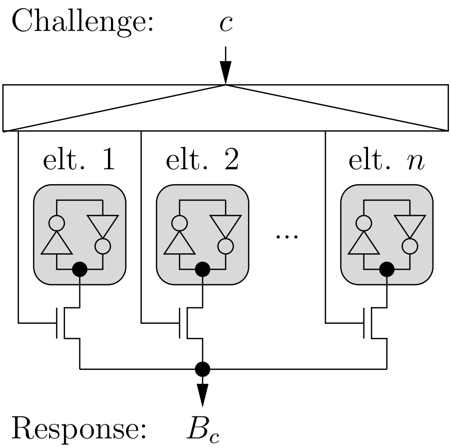
Convenorship: AENOR, Spain, Vice-convenorship: JISC, Japan

DOC TYPE: working draft

TITLE: **Text for ISO/IEC 1st WD 20897 — Information technology — Security requirements and test methods for physically unclonable functions for generating non-stored security parameters**

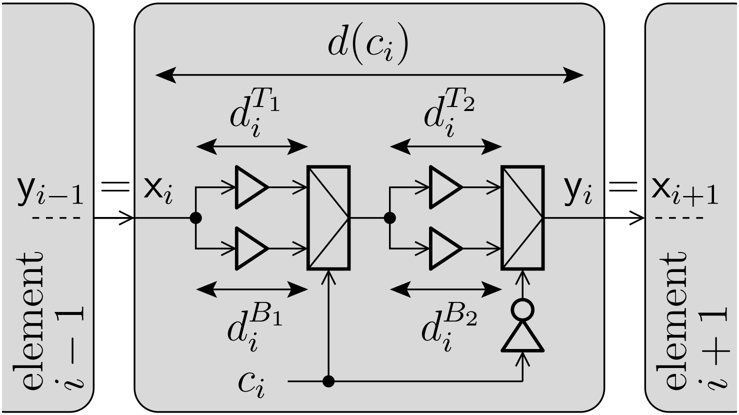
Outline

Non-delay PUF: SRAM PUF



Amount of entropy: $= n$.

Delay PUF: core delay element



Same idea as in other delay PUFs, like arbiter-PUF, etc.

Let $d(c_i)$ be the corresponding delay. As time is an extensive physical quantity:

$$d(c_i) = \begin{cases} d_i^{T_1} + d_i^{B_2} = d_i^{TB} & \text{if } c_i = -1, \\ d_i^{B_1} + d_i^{T_2} = d_i^{BT} & \text{if } c_i = +1. \end{cases}$$

The delays d_i^{TB} and d_i^{BT} are modeled as i.i.d. normal random variables selected at fabrication [PDW89].

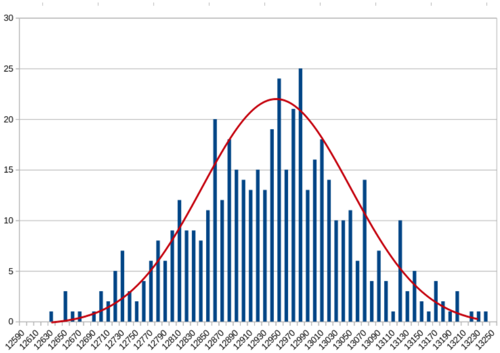
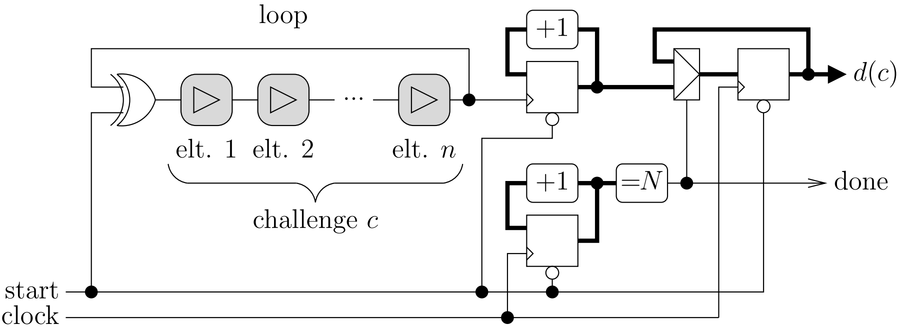


Figure : Monte-Carlo simulation (with 500 runs) of the delays in a chain of 60 basic buffers implemented in a 55 nm CMOS technology.

Delay PUF: Loop PUF

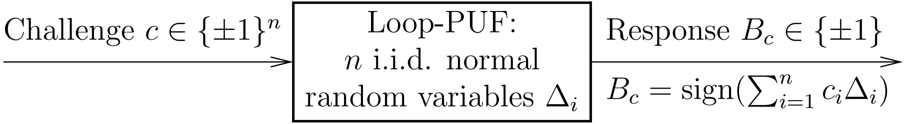


Amount of entropy: $> n$?

Nota bene: here, $d(c)$ is expressed in number of clock cycles.

LPUF is not self-contained

It needs a protocole



input : Challenge c

output: Response B_c

Measure $d_1 \leftarrow \lfloor N \sum_{i=1}^{n} d(c_i) \rfloor$

Set challenge $-c$

Measure $d_2 \leftarrow \lfloor N \sum_{i=1}^{n} d(-c_i) \rfloor$

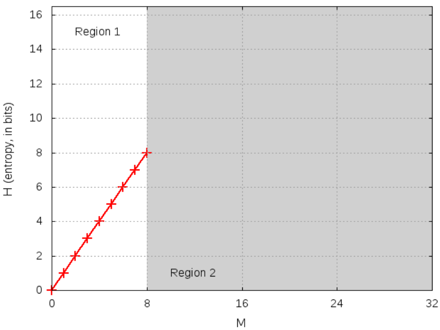
return $B_c = \mathrm{sign}(d_1 - d_2)$

Algorithm 1: Protocole to get one bit out LPUF.

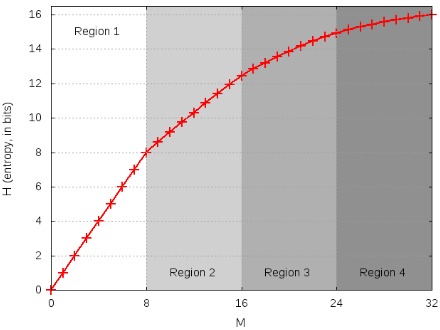
Our result: RAM-PUF vs Loop-PUF

For $n = 8$

SRAM-PUF



LPUF



Outline

Challenge

Definition

A *challenge* c is a vector of n control bits $c = (c_1, c_2, \ldots, c_n) \in \{\pm 1\}^n$. Let $\Delta_1, \Delta_2, \ldots, \Delta_n$ be i.i.d. zero-mean normal (Gaussian) variables characterizing the technological dispersion. A *bit response* to challenge c is defined as

$$B_c = \mathrm{sign}(\Delta_c) \in \{\pm 1\} \quad (1)$$

where

$$\Delta_c = c_1\Delta_1 + c_2\Delta_2 + \cdots + c_n\Delta_n. \quad (2)$$

Challenge code

Definition

A challenge *code* $\mathcal{C}$ is a set of M n-bit challenges that form a (n, M) binary code. We shall identify $\mathcal{C}$ with the $M \times n$ matrix of ± 1's whose lines are the challenges.
The M codewords and their complements are used to challenge the PUF elements. The corresponding identifier is the M-bit vector

$$B = (B_c)_{c \in \mathcal{C}}. \tag{3}$$

The *entropy* of the PUF responses is denoted by $H = H(B)$.

Orthant probabilities

Let $X_1, X_2, \ldots, X_n$ be zero-mean, jointly Gaussian (not necessarily independent) and identically distributed. As a prerequisite to the derivations that follow, we wish to compute the *orthant probability*

$$\mathbb{P}(X_1 > 0, X_2 > 0, \ldots, X_n > 0).$$

The probabilities associated to other sign combinations can easily be deduced from it using the symmetry properties of the Gaussian distribution.

Since the value of the orthant probability does not depend on the common variance of the random variables we may assume without loss of generality that each X_i has unit variance: $X_i \sim \mathcal{N}(0, 1)$. The orthant probability will depend only on the correlation coefficients

$$\rho_{i,j} = \mathbb{E}(X_i X_j) \qquad (i \neq j).$$

Some lemmas

Lemma (Quadrant probability of a bivariate normal)

$$\mathbb{P}(X_1 > 0, X_2 > 0) = \frac{1}{4} + \frac{\arcsin \rho_{1,2}}{2\pi}. \quad (5)$$

Lemma (Orthant probability of a trivariate normal)

$$\mathbb{P}(X_1 > 0, X_2 > 0, X_3 > 0) = \frac{1}{8} + \frac{\arcsin \rho_{1,2} + \arcsin \rho_{2,3} + \arcsin \rho_{1,3}}{4\pi}. \quad (6)$$

Lemma (No closed formula for $n > 3$ exists...)

Main Result: Hadamard Codes

We have M responses bits, so $H(B) \leq M$ bits.
When is it possible to have the maximum value $H(B) = M$ bits?

Main Result: Hadamard Codes

We have M responses bits, so $H(B) \leq M$ bits.

When is it possible to have the maximum value $H(B) = M$ bits?

Theorem

$H(B) = M$ implies $M \leq n$.

$H(B) = M = n$ bits if and only if $\mathcal{C}$ is a Hadamard (n, n) code.

Main Result: Hadamard Codes

We have M responses bits, so $H(B) \leq M$ bits.
When is it possible to have the maximum value $H(B) = M$ bits?

Theorem

$H(B) = M$ implies $M \leq n$.
$H(B) = M = n$ bits if and only if $\mathcal{C}$ is a Hadamard (n, n) code.

Proof.

$H(B) = M$ means that all bits B_c are independent, i.e., all $Y_j = \sum_{i=}^{n} c_i X_i$'s are independent (uncorrelated), i.e., all M (n-bit) challenges $c(j)$ are orthogonal. □

Hadamard Codes

n **orthogonal binary** ± 1 **vectors form an Hadamard code**:

- $n = 1$ $\mathcal{C} = (1)$, $H = 1$ bit;
- $n = 2$ $\mathcal{C} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$, $H = 2$ bits;
- $n = 3$ **No** Hadamard code! but any (3,3) code $\equiv \begin{pmatrix} 1 & 1 & 1 \\ -1 & 1 & 1 \\ 1 & -1 & 1 \end{pmatrix}$ for which $\Sigma = \frac{1}{3}\mathcal{C}\mathcal{C}^t = \begin{pmatrix} 1 & 1/3 & 1/3 \\ 1/3 & 1 & -1/3 \\ 1/3 & -1/3 & 1 \end{pmatrix}$ gives

$$\begin{aligned} H = & -6\left(\tfrac{1}{8} + \tfrac{\arcsin 1/3}{4\pi}\right) \log\left(\tfrac{1}{8} + \tfrac{\arcsin 1/3}{4\pi}\right) \\ & - 2\left(\tfrac{1}{8} - 3\tfrac{\arcsin 1/3}{4\pi}\right) \log\left(\tfrac{1}{8} - 3\tfrac{\arcsin 1/3}{4\pi}\right) \\ \approx & \; 2.875 < 3 \text{ bits.} \end{aligned}$$

Hadamard Codes (cont'd)

n=4 $$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}, H = 4 \text{ bits}$$

n=8 $$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{pmatrix}, H = 8 \text{ bits}$$

Hadamard Codes (cont'd)

n=12

$$\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 \\
1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 \\
1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 \\
1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 \\
1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 \\
1 & -1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & 1 \\
1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 \\
1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 \\
1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 \\
1 & -1 & 1 & -1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1
\end{pmatrix}$$

$H = 12$ bits

Outline

Beyond n bits

$n = 1$ element $\implies H = 1$ bit;

Beyond n bits

$n = 1$ element $\implies H = 1$ bit;
$n = 2$ elements $\implies H = 2$ bits;

Beyond n bits

$n = 1$ element $\implies H = 1$ bit;

$n = 2$ elements $\implies H = 2$ bits;

$H = M$ (max entropy = number of challenges) $\implies H \leq n$ bits.

Beyond n bits

$n = 1$ element $\implies H = 1$ bit;

$n = 2$ elements $\implies H = 2$ bits;

$H = M$ (max entropy = number of challenges) $\implies H \leq n$ bits.

Common belief that n elements give at most n bits of entropy (SRAM PUFs, delay PUFs).

Q Can we obtain more than n bits by taking more challenges: $M > n$?

Beyond n bits

$n = 1$ element $\implies H = 1$ bit;

$n = 2$ elements $\implies H = 2$ bits;

$H = M$ (max entropy = number of challenges) $\implies H \leq n$ bits.

Common belief that n elements give at most n bits of entropy (SRAM PUFs, delay PUFs).

Q Can we obtain more than n bits by taking more challenges: $M > n$?

A Yes!

$$\boxed{n < H < M}$$

For n elements, using $M > n$ challenges, the entropy can increase beyond n bits, albeit strictly $< M$.

$n = 3$ elements

$M = 1$ $\mathcal{C}_1 = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$ gives $H = 1$ bit.

$n = 3$ elements

$M = 1$ $\mathcal{C}_1 = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$ gives $H = 1$ bit.

$M = 2$ $\mathcal{C}_2 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \end{pmatrix}$ gives $H =$

$$-\left(\tfrac{1}{2} + \tfrac{\arcsin 1/3}{\pi}\right) \log\left(\tfrac{1}{4} + \tfrac{\arcsin 1/3}{2\pi}\right) - \left(\tfrac{1}{2} - \tfrac{\arcsin 1/3}{\pi}\right) \log\left(\tfrac{1}{4} - \tfrac{\arcsin 1/3}{2\pi}\right) \approx 1.966 \text{ bits.}$$

$n = 3$ elements

$M = 1$ $\mathcal{C}_1 = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$ gives $H = 1$ bit.

$M = 2$ $\mathcal{C}_2 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \end{pmatrix}$ gives $H =$

$$-\left(\tfrac{1}{2} + \tfrac{\arcsin 1/3}{\pi}\right) \log\left(\tfrac{1}{4} + \tfrac{\arcsin 1/3}{2\pi}\right) - \left(\tfrac{1}{2} - \tfrac{\arcsin 1/3}{\pi}\right) \log\left(\tfrac{1}{4} - \tfrac{\arcsin 1/3}{2\pi}\right) \approx 1.966 \text{ bits.}$$

$M = 3$ $\mathcal{C}_3 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & 1 \end{pmatrix}$ gives $H =$

$$-\left(\tfrac{3}{4} + 3\tfrac{\arcsin 1/3}{2\pi}\right) \log\left(\tfrac{1}{8} + \tfrac{\arcsin 1/3}{4\pi}\right) - \left(\tfrac{1}{4} - 3\tfrac{\arcsin 1/3}{2\pi}\right) \log\left(\tfrac{1}{8} - 3\tfrac{\arcsin 1/3}{4\pi}\right) \approx 2.875 \text{ bits.}$$

$n = 3$ elements

$M = 1$ $\mathcal{C}_1 = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$ gives $H = 1$ bit.

$M = 2$ $\mathcal{C}_2 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \end{pmatrix}$ gives $H =$

$$-\left(\frac{1}{2} + \frac{\arcsin 1/3}{\pi}\right) \log\left(\frac{1}{4} + \frac{\arcsin 1/3}{2\pi}\right) - \left(\frac{1}{2} - \frac{\arcsin 1/3}{\pi}\right) \log\left(\frac{1}{4} - \frac{\arcsin 1/3}{2\pi}\right) \approx 1.966 \text{ bits.}$$

$M = 3$ $\mathcal{C}_3 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & 1 \end{pmatrix}$ gives $H =$

$$-\left(\frac{3}{4} + 3\frac{\arcsin 1/3}{2\pi}\right) \log\left(\frac{1}{8} + \frac{\arcsin 1/3}{4\pi}\right) - \left(\frac{1}{4} - 3\frac{\arcsin 1/3}{2\pi}\right) \log\left(\frac{1}{8} - 3\frac{\arcsin 1/3}{4\pi}\right) \approx 2.875 \text{ bits.}$$

$M = 4$ $\mathcal{C}_3 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & 1 \\ -1 & 1 & 1 \end{pmatrix}$ gives ≈ 3.666 bits

$n = 3$ elements

$M = 1$ $\mathcal{C}_1 = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$ gives $H = 1$ bit.

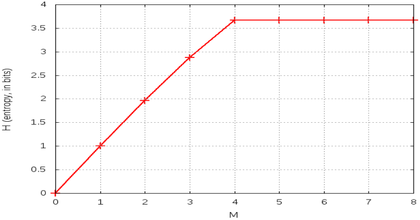
$M = 2$ $\mathcal{C}_2 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \end{pmatrix}$ gives $H =$

$$-\left(\tfrac{1}{2} + \tfrac{\arcsin 1/3}{\pi}\right) \log\left(\tfrac{1}{4} + \tfrac{\arcsin 1/3}{2\pi}\right) - \left(\tfrac{1}{2} - \tfrac{\arcsin 1/3}{\pi}\right) \log\left(\tfrac{1}{4} - \tfrac{\arcsin 1/3}{2\pi}\right) \approx 1.966 \text{ bits.}$$

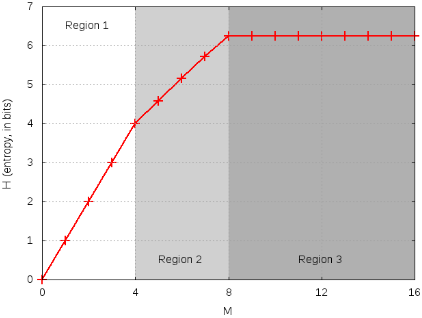
$M = 3$ $\mathcal{C}_3 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & 1 \end{pmatrix}$ gives $H =$

$$-\left(\tfrac{3}{4} + 3\tfrac{\arcsin 1/3}{2\pi}\right) \log\left(\tfrac{1}{8} + \tfrac{\arcsin 1/3}{4\pi}\right) - \left(\tfrac{1}{4} - 3\tfrac{\arcsin 1/3}{2\pi}\right) \log\left(\tfrac{1}{8} - 3\tfrac{\arcsin 1/3}{4\pi}\right) \approx 2.875 \text{ bits.}$$

$M = 4$ $\mathcal{C}_3 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & 1 \\ -1 & 1 & 1 \end{pmatrix}$ gives ≈ 3.666 bits



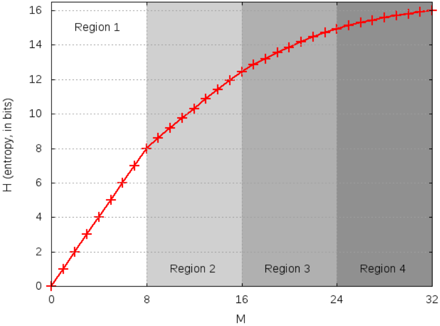
$n = 4$ elements



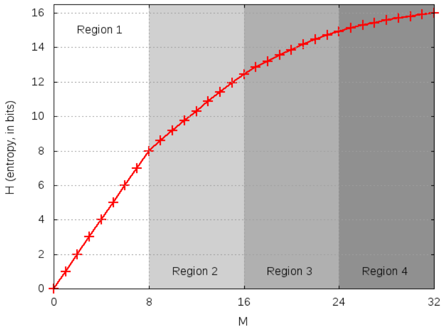
$$\mathcal{C}_8 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{pmatrix}$$

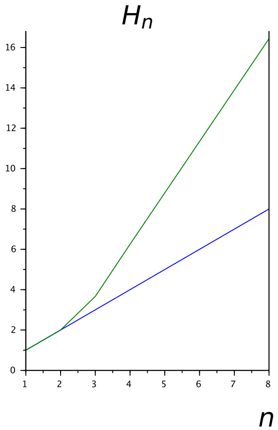
$H = 6.251$ bits.

$n = 8$ elements, etc.



$n = 8$ elements, etc.





Outline

Conclusions and Perspectives

Conclusions

- $H_n = n$ bits of entropy obtained using a Hadamard challenge code;
- $H_n > n$ bits of entropy obtained using a challenge code made of several Hadamard “chunks”

Related talk given at ISIT 2016 [RSGD16]:

[CDGB12] Zouha Cherif, Jean-Luc Danger, Sylvain Guilley, and Lilian Bossuet.
An easy-to-design PUF based on a single oscillator: The loop PUF.
In *15th Euromicro Conference on Digital System Design, DSD 2012, Çeşme, Izmir, Turkey, September 5-8, 2012*, pages 156–162. IEEE Computer Society, 2012.

[PDW89] Marcel J.M. Pelgrom, Aad C.J. Duinmaijer, and Anton P.G. Welbers.
Matching properties of MOS transistors.
IEEE Journal of Solid State Circuits, 24(5):1433–1439, 1989.
DOI: 10.1109/JSSC.1989.572629.

[RSGD16] Olivier Rioul, Patrick Solé, Sylvain Guilley, and Jean-Luc Danger.
On the Entropy of Physically Unclonable Functions.
In *ISIT, IEEE International Symposium on Information Theory*, July 2016.
Barcelona, Spain.