Une école de l'IMT

The Big Picture of Delay-PUF Dependability

ECCTD, Sofia, September 9, 2020

Alexander Schaub[a], Jean-Luc Danger[ab], Sylvain Guilley[ab], Olivier Rioul[ac]

[a]LTCI, Télécom ParisTech, France
[b]Secure-IC, France
[c]CMAP, Ecole Polytechnique, France

Physically Unclonable Functions
Motivation

- Embedded Security
- Anti-counterfeiting
- Secure password storage

175,000 IoT cameras can be remotely hacked thanks to flaw, says security researcher

Researchers have found that it's trivial to remotely access one brand of security camera.

Nintendo's Switch can be hacked to run custom apps and games

Home > Access Control > Passwords

INSIDER FEATURE

1.4B stolen passwords are free for the taking: What we know now

The 2012 LinkedIn password breach, and others like it, are still paying dividends for criminals

Contents

Table of Contents

Physically Unclonable Functions (PUFs)
PUF Definition

Definition

A **PUF** is a physical device defined by:

- Input: challenge bit-string $c \in \{0,1\}^n$,
- Output: bit response $\mathcal{P}(c) \in \{0,1\}$.

The mapping is *random* and different for each device, which should not be clonable (physically and mathematically).

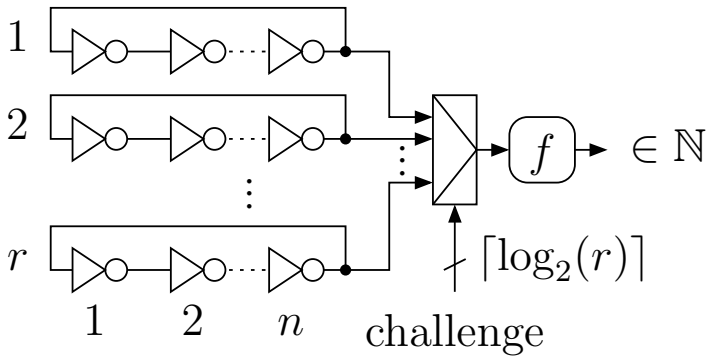
Note

Real PUFs are not deterministic and subject to

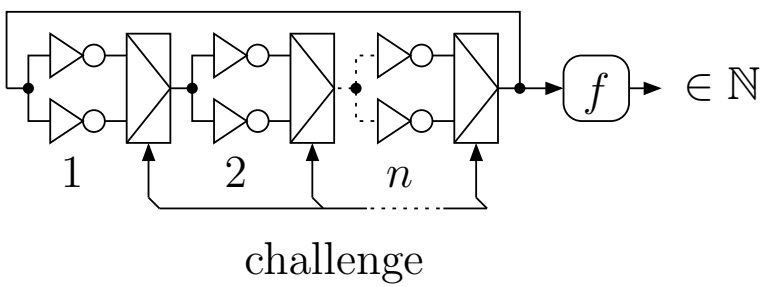
- noise
- aging
- etc.

Types of Unclonable Functions
Zoom on Delay PUFs

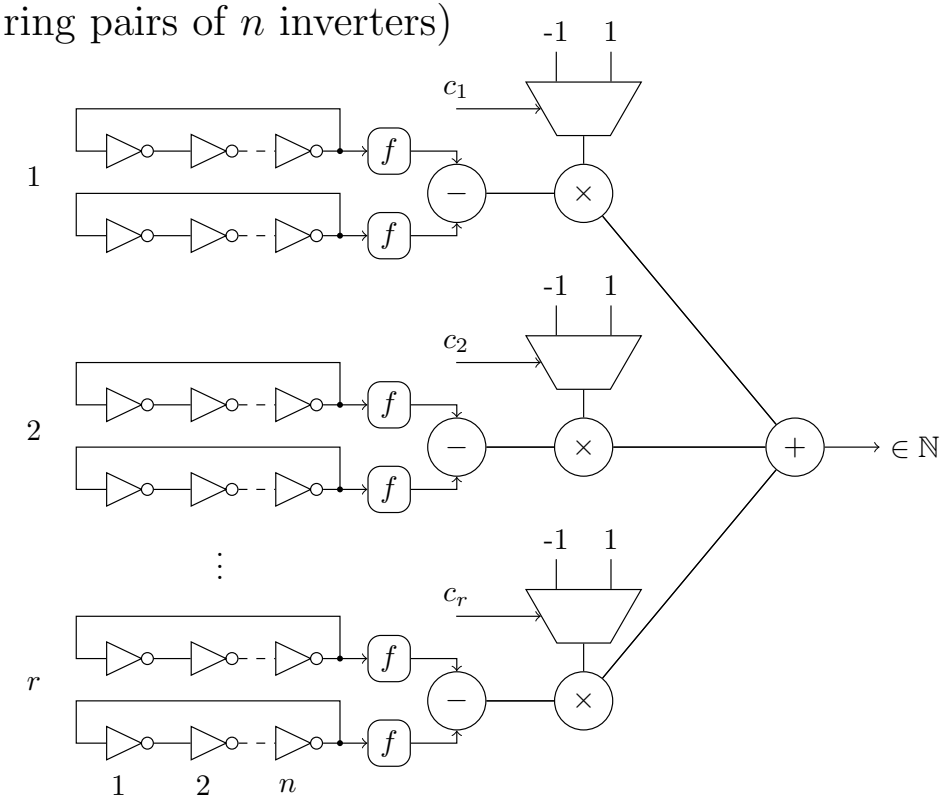
Ring-oscillator PUF
(r rings of n inverters)



Loop-PUF
(with n inverters)



RO-sum PUF
(r ring pairs of n inverters)



Modelisation of Delay PUFs

Ideal (noiseless) output: $\mathcal{P}(c) = \text{sign}(c \cdot x)$ where $x \in \mathbb{R}^n$ represents the internal PUF state. For a PUF subject to additive noise Z:

$$\mathcal{P}(c) = \text{sign}(c \cdot x + Z)$$

where:

- Static (manufacturing) randomness: $x \sim \mathcal{N}(0, \Sigma^2)$
- Dynamic (measurement) randomness: $Z \sim \mathcal{N}(0, \sigma^2)$

We define the Signal-to-Noise Ratio as:

$$SNR = \frac{n\Sigma^2}{\sigma^2}$$

Table of Contents

Reliability

Independency Hypothesis

In this section, we suppose that $\mathcal{P}(c)$ are independent for different challenges c (can be achieved by restricting the challenge space to a size $\leq n$).

This section will present results about how the reliability of PUFs can be improved – how the dynamic randomness can be reduced.

Reliability

BER: Definition and Expression

Definition

BER (bit error rate): probability that the outcome differs from the ideal output:

$$\mathrm{BER} = \mathbb{P}_Z[\mathsf{sign}(c \cdot x + Z) \neq \mathsf{sign}(c \cdot x)]$$

Averaged over the static randomness:

$$\widehat{\mathrm{BER}} = \mathbb{E}_{c \cdot x}[\mathrm{BER}]$$

Theorem (Closed Form Expressions for BER)

$$\mathrm{BER} = \frac{1}{2}\,\mathsf{erfc}\left(\frac{|c \cdot x|}{\sigma\sqrt{2}}\right)$$

$$\widehat{\mathrm{BER}} = \frac{1}{\pi}\,\arctan\left(\frac{1}{\sqrt{\mathrm{SNR}}}\right)$$

BER Expression Proofs

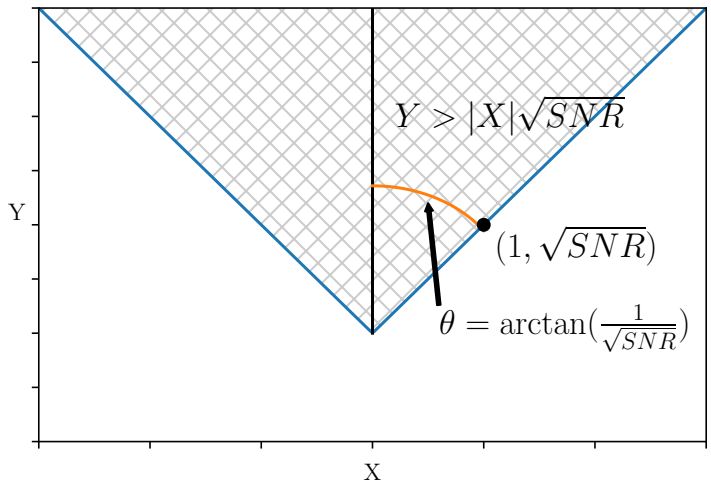
Proof (BER).

Assume $c \cdot x > 0$ (symmetrical proof for $c \cdot x < 0$).

$$\mathbb{P}[c \cdot x + Z < 0] = \mathbb{P}[Z < -c \cdot x] = \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{-c \cdot x} e^{\frac{-z^2}{2\sigma^2}} dz = \frac{1}{2} \operatorname{erfc}\left(\frac{c \cdot x}{\sigma\sqrt{2}}\right) \square$$

Proof (Averaged BER).

$\mathbb{P}[\text{sign}(c \cdot x + Z) \neq \text{sign}(c \cdot x)] = \mathbb{P}[Z > |c \cdot x|]$.
But
$Z > |c \cdot x| \Leftrightarrow \frac{Z}{\sigma} > \frac{|c \cdot x|}{\Sigma} \frac{\Sigma}{\sigma} \Leftrightarrow Y > |X|\sqrt{SNR}$
where $X, Y \sim \mathcal{N}(0, 1)$ $\square$



Reliability Enhancement

- Usually, the reliability achieved by raw PUFs is not enough for the desired applications.
- Reliability can be improved via different means:
  - Filtering out unreliable responses (either a fixed number of responses, or all responses less reliable than a given threshold)
  - Using error correcting codes
  - (Novel Two-Metric Helper Data [IWASI 2019])

$\Longrightarrow$ which method is more effective for delay-PUFs ?

Table of Contents

Reliability Improvement: Filtering

- Method chosen: discard all "unreliable" bits
- Unreliable: $|c \cdot x| < \Theta\sigma$ for some chosen $\Theta > 0$.
- Tradeoff: improves reliability but reduces number of output bits
- Discarding a fixed number of challenges: also possible but more complicated to analyze.

Reliability Enhancement: Bit filtering

Theorem (Average BER After Filtering)

$$\widehat{\mathrm{BER}}_{filt} = \frac{2}{\mathrm{erfc}\left(\frac{\Theta}{\sqrt{2}\cdot\sqrt{\mathrm{SNR}}}\right)}\left(T\left(\Theta, \frac{1}{\sqrt{\mathrm{SNR}}}\right) + \frac{1}{4}\,\mathrm{erf}\left(\frac{\Theta}{\sqrt{2}\cdot\sqrt{\mathrm{SNR}}}\right)\left(\mathrm{erf}\left(\frac{\Theta}{\sqrt{2}}\right) - 1\right)\right)$$

where T is Owen's T-function:

$$T(h, a) = \frac{1}{2\pi}\int_0^a \frac{e^{-\frac{1}{2}h^2(1+x^2)}}{1+x^2}\,\mathrm{d}x.$$

Reliability Enhancement: Bit filtering
Continued

Theorem (Average Entropy After Filtering)

Average (over static randomness) entropy for n delay elements:

$$\widehat{H(n,\Theta)}_{SNR} = n \cdot \mathrm{erfc}\left(\frac{\Theta}{\sqrt{2\mathrm{SNR}}}\right).$$

Reliability Enhancement: Fuzzy Extraction

During enrollment:

1. generate PUF string $b = (\text{sign}(c \cdot x_1), \ldots, \text{sign}(c \cdot x_n))$
2. generate a random codeword w as a reliable identifier to compute the public helper data $d = w \oplus b$;
3. store the public helper data d in the PUF.

During bitstring generation:

1. a PUF (unreliable) measurement is made, yielding b';
2. the measurement is XOR'd to the public helper data to obtain a noisy codeword $w' = d \oplus b'$;
3. the noisy codeword w' is decoded to obtain w.

Reliability Enhancement: Fuzzy Extraction

We suppose that a $[n, k, d]_2$ code is used for error correction.

Theorem (Entropy After Fuzzy Extraction)

The remaining entropy is equal to

$$H(n) = k.$$

Theorem (Key Error Rate After Fuzzy Extraction)

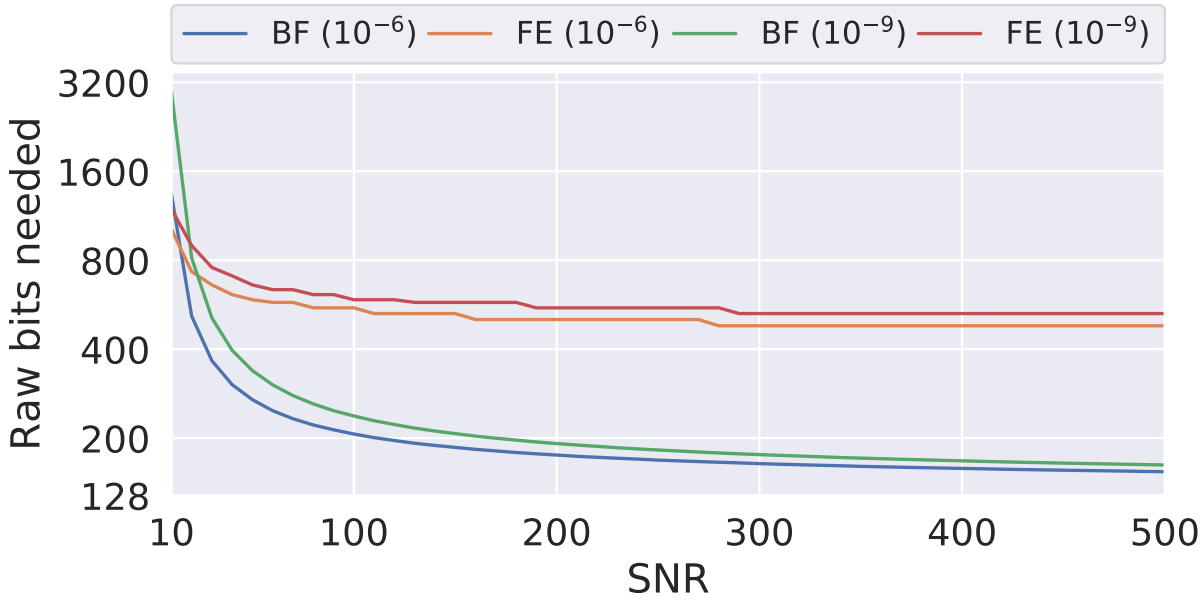
$$\mathrm{KER}(n) \leq 1 - \mathbb{P}\{\mathcal{B}(n, p) \leq d\}$$

where $p = \widehat{BER}(n)$ *and* $\mathcal{B}(n, p)$ *is the* (n, p)*-binomial distribution.*

Fuzzy Extraction in Practice

- The decoding step should allow an efficient hardware implementation.
- Not all codes can be used !
- One possibility (used by PUFKY):
  - An efficiently decodable outer code in $GF(2^q)$: Reed-Muller, Hadamard or Golay code, and
  - A short inner repetition code
- This solution is efficiently decodable and gives a better bound on the key error rate.

Reliability Enhancement: Comparison



This figure represents the raw bits needed to obtain a 128-bit key with KER 10^{-6} and 10^{-9} using fuzzy extraction (FE) or bit filtering (BF). For reasonable values of the SNR, bit filtering is far more efficient than fuzzy extraction. For very low SNR, fuzzy extraction degrades more gracefully than bit filtering.

Table of Contents

Correlated Responses

Linear Correlation

In this section we will suppose that the values $c \cdot x$ are linearly correlated for different challenges c.

A PUF of size n admits up to 2^n different (correlated) challenges. The max-entropy converges towards n^2 for large n. Simulations suggest a similar behavior for the Shannon entropy.

In contrast, only an entropy of up to n can be achieved when limited to independent challenges.

Entropy after Filtering Correlated Challenges

Exact values could only be obtained for two correlated challenges.

Theorem (Entropy After Correlation)

Let c_1, c_2 two challenges, $p_1 = \mathbb{P}\{c_1 \cdot x > 0, c_2 \cdot x > 0 \mid |c_1 \cdot x|, |c_2 \cdot x| > \Theta\sigma\}$, $p_2 = \mathbb{P}\{c_1 \cdot x > 0, c_2 \cdot x < 0 \mid |c_1 \cdot x|, |c_2 \cdot x| > \Theta\sigma\}$. Then

$$\frac{p_1}{p_2} = \frac{\frac{1}{2} - \operatorname{erf}\left(\frac{\Theta}{\sqrt{2}\cdot\sqrt{\mathrm{SNR}}}\right)/2 - 2T\left(\frac{\Theta}{\sqrt{\mathrm{SNR}}}, \sqrt{\frac{n_2}{n_1}}\right)}{\frac{1}{2} - \operatorname{erf}\left(\frac{\Theta}{\sqrt{2}\cdot\sqrt{\mathrm{SNR}}}\right)/2 - 2T\left(\frac{\Theta}{\sqrt{\mathrm{SNR}}}, \sqrt{\frac{n_1}{n_2}}\right)}$$

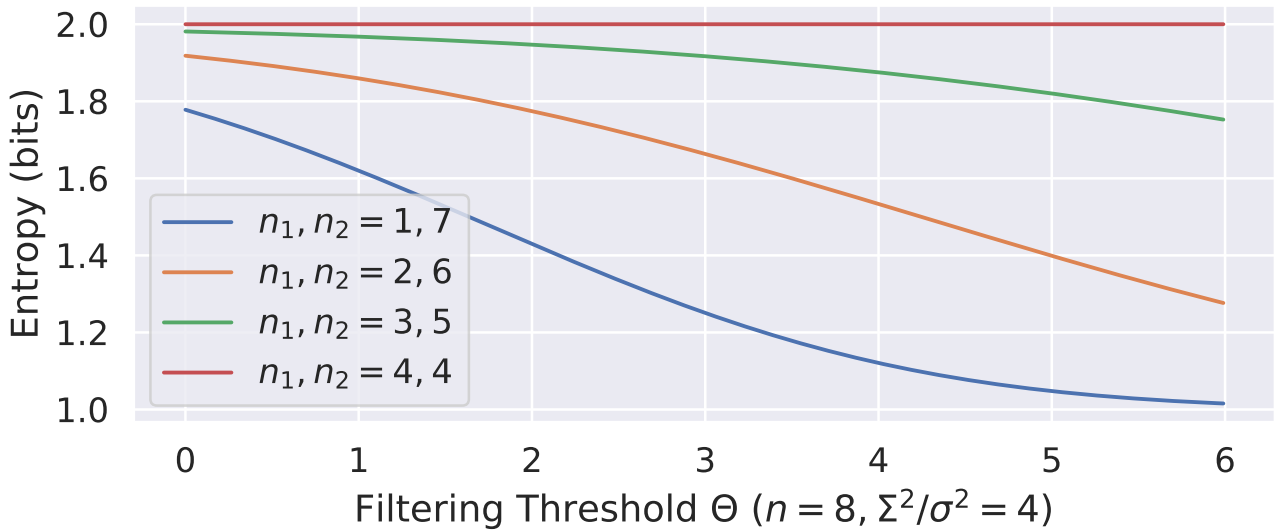
where $n_1 = \frac{n+c_1\cdot c_2}{2}$, $n_2 = \frac{n-c_1\cdot c_2}{2}$ and T is Owen's T-function.

This formula gives the ratio of the probability that two challenges have the same sign divided by the probability that have different signs, assuming both are reliable with threshold Θ.

Entropy after Filtering Correlated Challenges

Example for $n = 8$

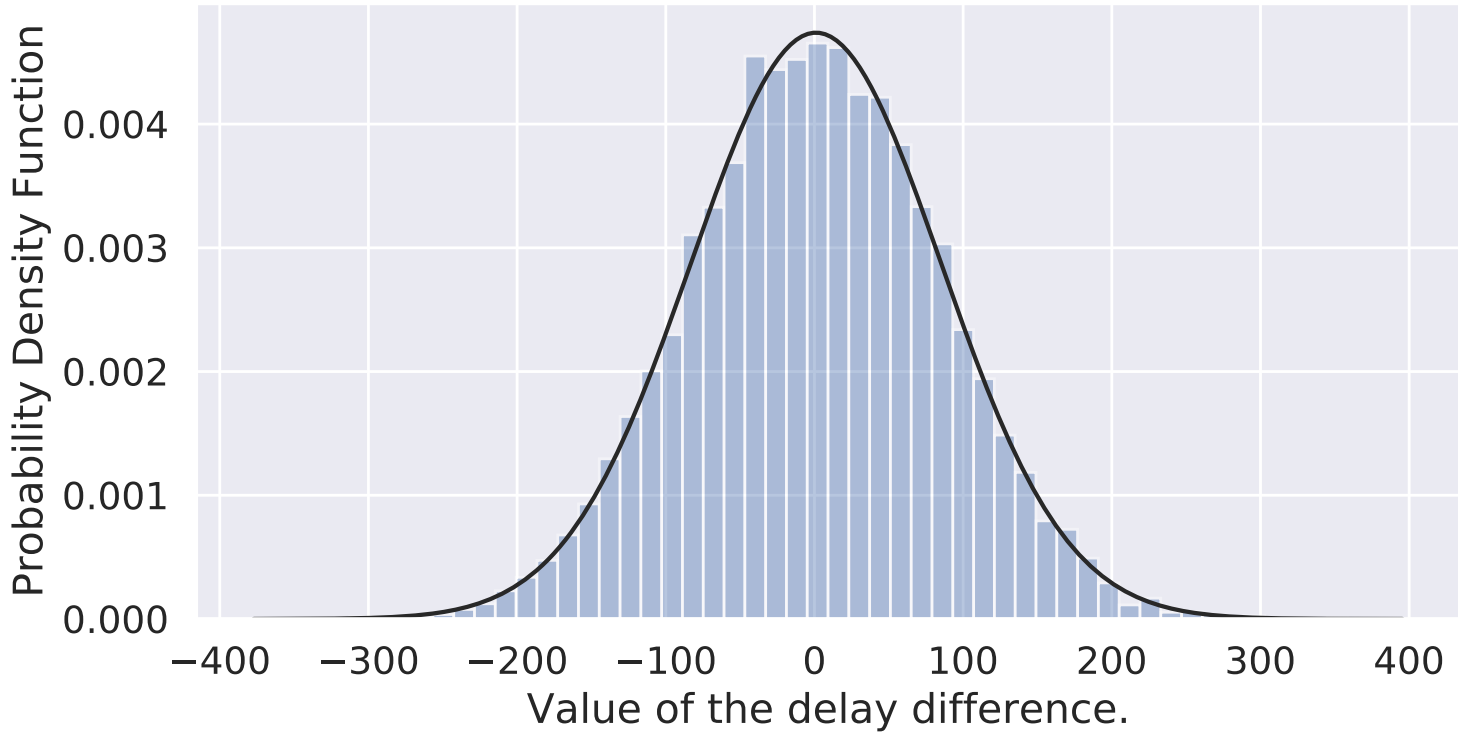
Entropy = that of a distribution with 4 outcomes of probabilities p_1, p_1, p_2 and p_2. Let us denote $p1/p2 = c$. We have $p1 + p2 = 1/2$, hence $p_2 = 1/(2(1+c))$, $p_1 = c/(2(1+c))$, and entropy is equal to $1 + h(1/(1+c))$.



For strongly correlated challenges, although there is a high entropy without filtering, it degrades rapidly into low entropy when filtering. This limits the usefulness of using highly correlated challenges for improving entropy.

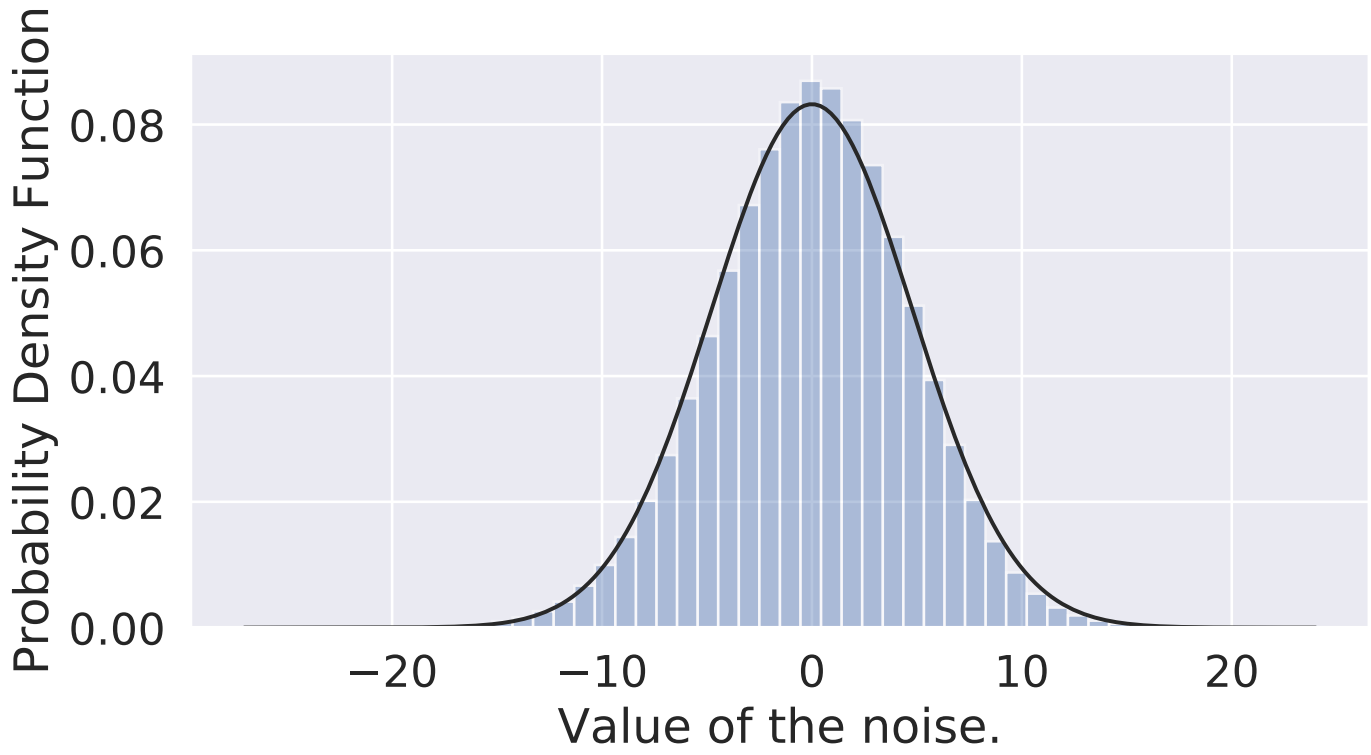
Table of Contents

Gaussian PUF Distribution



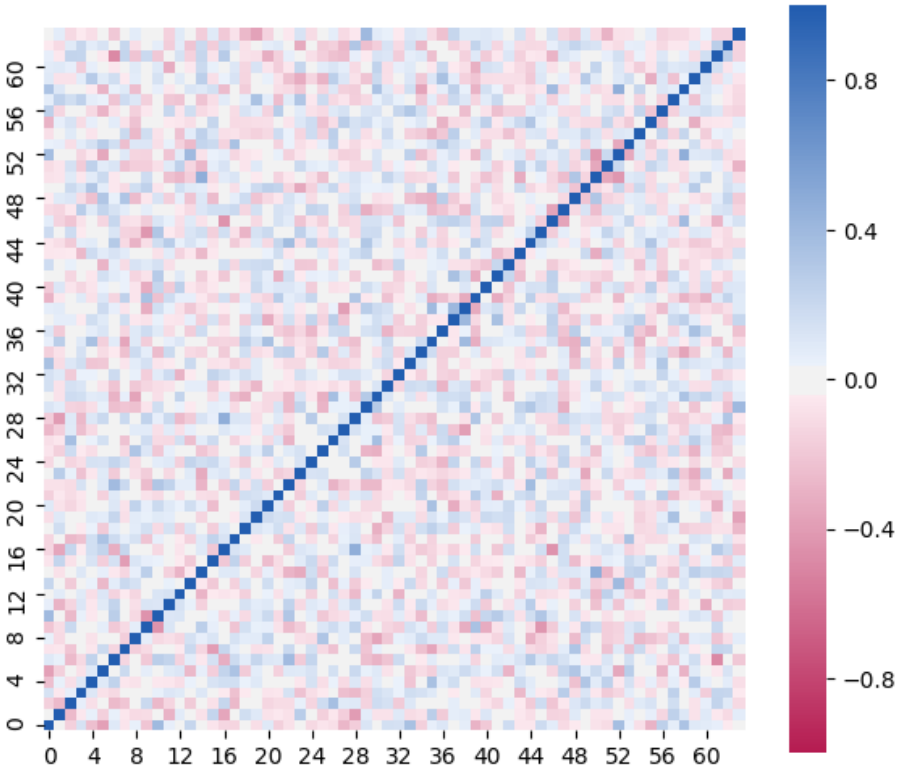
The empiric distribution of several hundred PUFs with size $n = 64$ has been computed. This validates the Gaussian model for the PUF delay distribution ($n\Sigma^2 \simeq 7086$).

Gaussian Noise Distribution



The empiric distribution of the noise has been computed after repeated measurements. This validates the Gaussian model for the PUF noise distribution ($\sigma^2 \simeq 23$).

Delay Decorrelation Hypothesis



We assumed the internal state of the PUF $x \in \mathbb{R}^n$ was distributed as an n independent Gaussian variables.This can be verified through the distribution of the linear correlation coefficients. After rescaling, uniformity hypothesis was performed, yielding a p-value of $0.158 > 0.05$ and validating the independency hypothesis.

Wrap up

- A comparison of different reliability enhancing techniques: filtering vs fuzzy extraction
- Tradeoff characterized in terms of bit error rate and entropy
- Filtering under correlation: first results quantify entropy loss
- Post-silicon testing validates the model

Future directions

- More than two dependent challenges ?
- Show a linear growth of Shannon entropy with n...
- Noise does not seem to be strictly independent: a more refined modelization is needed.

For industrial deployments:

- `https://www.secure-ic.com/physically-unclonable-function/`
- `https://www.secure-ic.com/white-paper-download-puf/`

Une école de l'IMT

The Big Picture of Delay-PUF Dependability

ECCTD, Sofia, September 9, 2020

Alexander Schaub[a], Jean-Luc Danger[ab], Sylvain Guilley[ab], Olivier Rioul[ac]

[a]LTCI, Télécom ParisTech, France
[b]Secure-IC, France
[c]CMAP, Ecole Polytechnique, France