HAL open science

Challenge codes for physically unclonable functions with Gaussian delays: A maximum entropy problem

Alexander Schaub, Olivier Rioul, Joseph Boutros, Jean-Luc Danger, Sylvain Guilley

▶ **To cite this version:**

Alexander Schaub, Olivier Rioul, Joseph Boutros, Jean-Luc Danger, Sylvain Guilley. Challenge codes for physically unclonable functions with Gaussian delays: A maximum entropy problem. Latin American Workshop on Coding and Information, Jul 2018, Unicamp-Campinas, Brazil. hal-02287965

HAL Id: hal-02287965

https://telecom-paris.hal.science/hal-02287965v1

Submitted on 12 Aug 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Challenge Codes for Physically Unclonable Functions with Gaussian Delays: A Maximum Entropy Problem

Alexander Schaub*, Olivier Rioul*†, Joseph J. Boutros‡, Jean-Luc Danger*§ and Sylvain Guilley*§
* LTCI, Telecom ParisTech, 75013 Paris, France
firstname.lastname@telecom-paristech.fr
† CMAP, Ecole Polytechnique, 91120 Palaiseau, France
olivier.rioul@polytechnique.edu
‡ Texas A&M University, 23874 Doha, Qatar
boutros@tamu.edu
§ Secure-IC S.A.S.
35510 Cesson-Sévigné, France

I. Introduction and motivation

Suppose we are given a (nonlinear) (n, M) code C with M codewords $c_i \in \{\pm 1\}^n$ and n i.i.d. standard Gaussian variables $X_1, X_2, \ldots, X_n \sim \mathcal{N}(0, 1)$. Consider the scalar products

$$c_i \cdot X = \sum_{j=1}^{n} c_{i,j} X_j \qquad (i = 1, 2, \ldots, M)$$

where $X = (X_1, X_2, \ldots, X_n)$ and the associated sign bits

$$B_i = \operatorname{sgn}(c_i \cdot X) \in \{\pm 1\} \qquad (i = 1, 2, \ldots, M).$$

The question addressed in this paper is the following. What is the joint entropy of the sign bits

$$H_C \stackrel{\text{def.}}{=} H(B_1, B_2, \ldots, B_M) \quad ?$$

In particular, can we evaluate the *maximum entropy* $H_n = \max_C H_C$ attained for the full universe code $C = \{\pm 1\}^n$? Despite appearances, this problem turns out to be purely combinatorial as shown below.

The motivation for this problem comes from hardware security. Modern secure integrated circuits make use of hardware primitives called *physically unclonable functions* (PUFs) that can generate unique identifiers from challenges, such as described, for example, by Maes [2]. Having a unique identifier for each physical chip allows one to authenticate it in a secure way. PUFs exploit small, uncontrollable physical variations of the manufacturing process that cannot be replicated, hence the name "physically unclonable". Such small variations are often modeled by Gaussian random variables.

The PUF is a function that takes several challenges $c_1, c_2, \ldots, c_M$ (the so-called *challenge code*) as inputs and returns the bitvector identifier $(B_1, B_2, \ldots, B_M)$. The definitions above correspond to a particular PUF that exploits the variability of n distinct delay elements (a so-called delay-PUF referred to as a "Loop PUF"), where $X_1, X_2, \ldots, X_n$ are independent *Gaussian delay* differences. This PUF is described in detail by Cherif et al. [1] while the mathematical model is explored in depth by Rioul et al. [3]. To assess the security of a PUF, it is necessary that the entropy of the identifier's distribution $H_C = H(B_1, B_2, \ldots, B_M)$ is sufficiently high. For a given value of M, the optimal challenge code C maximizes H_C. Since adding more challenges can only increase the entropy, it is important to evaluate the maximum possible value H_n (for $M = 2^n$) as a function of n. In fact, H_n is achieved with codes of size $M = 2^{n-1}$ since $\{\pm 1\}^n$ can be partititioned into two opposite sets where codewords in the second set bring no entropy.

Rioul et al. [3] showed that the optimal challenge code when $M \leq n$ is given by a Hadamard code[1] C for which one can attain $H_C = n$ bits of identification. In general, we have $H_n > n$ and numerical simulations suggest that H_n becomes much greater than n (despite a common belief in hardware security that the entropy would be limited

[1]When such a Hadamard code exists, which implies that $n = 1, 2$ or a multiple of 4.

by n, the number of elements in a PUF). To assess the security of the PUF, it is important to evaluate how the entropy H_n grows as the complexity of the PUF, n, increases.

The exact calculation of H_C or H_n can be carried out only for very small values of n. Rioul et al. [3] give the exact values of H_C for $n, M \leq 3$ using well-known closed-form formulas for orthant probabilities of bi- and tri-variate normals. We give below a combinatorial extension that provides the exact values of H_C for $n = 3, 4$ even though the corresponding closed-form orthant probability formula is not available. For $n \geq 5$, the method soon becomes intractable and one has recourse to numerical computations.

The value of the maximum entropy H_n can be estimated reliably by defining equivalence classes of challenge codewords corresponding to the same value of joint probabilities

$$\mathbb{P}_b = \mathbb{P}_{b_1,\ldots,b_M} = \mathbb{P}\{B_1 = b_1, B_2 = b_2, \ldots, B_M = b_M\}.$$

In this paper, we derive a method that determines all such equivalent classes. Perhaps surprisingly, this problem is purely of discrete combinatorial nature. The actual values of the corresponding probabilities (hence that of H_n) are then estimated by Monte Carlo simulation. This method was found to be numerically tractable for $n = 5, 6, 7$.

We show that for $n \geq 4$ the relative number of zero probabilities approaches the total number of probabilities, i.e., $|\{b : \mathbb{P}_b > 0\}|/2^{2^{n-1}} \to 0$. We determine the exact number of zero probabilities for $n = 4, 5, 6, 7, 8$. This provides the exact value of the max-entropy, a reasonably tight upper bound of H_n.

For even greater values of n we propose a method inspired by compressed sensing using a randomly chosen sparse matrix S and change the code C to SC. Preliminary results are shown in Section V.

It is interesting to note that other applications of this type of entropy maximization problem can be considered. The PUF itself can be used to generate M-bit cryptographic keys. In another direction, one can think of the design of a true random number generator (TRNG) that combines n noisy measurements X using the code C represented by an $M \times n$ matrix $C = ((c_{i,j}))_{i,j}$ to obtain a M-dimensional vector CX. In this way the resulting M random sign bits are equiprobable and have joint entropy H_C much greater than the number of measurements n. Yet another application uses transform source coding with 1-bit scalar quantization (sign bit computation) applied to a transformed Gaussian memoryless source with a linear transformation C (such as a Walsh-Hadamard transform) e.g., for classification purposes.

II. Closed-Form Expressions

In the sequel, $b = (b_1 b_2 \ldots b_M)$ is called a sign vector.

A. Case $n = 3$

By considering the challenge matrix $C_3 = \left(\begin{smallmatrix} 1 & 1 & 1 \\ 1 & 1 & - \\ 1 & - & 1 \end{smallmatrix}\right)$, exact probabilities $\mathbb{P}_b$ can be derived by using the formula for tri-variate Gaussian distributions described by Rioul et al. [3]. This yields an entropy of

$$H_{C_3} = -\Big(\frac{1}{4} - 3\frac{\arcsin\frac{1}{3}}{2\pi}\Big)\log\Big(\frac{1}{8} - 3\frac{\arcsin\frac{1}{3}}{4\pi}\Big).$$

For the matrix with four challenges $C_4 = \left(\begin{smallmatrix} 1 & 1 & 1 \\ 1 & 1 & - \\ 1 & - & 1 \\ - & 1 & 1 \end{smallmatrix}\right)$ and the two sign vectors $+---$ and $-+++$, we have that

$$\mathbb{P}_{+---} = \mathbb{P}_{-+++} = 0 \text{ (see Section III)}$$

By exploiting symmetries, it follows that eight sign vectors satisfy

$$\mathbb{P}_{++++} = \mathbb{P}_{++--} = \mathbb{P}_{+-+-} = \mathbb{P}_{+--+} = \mathbb{P}_{----} = \mathbb{P}_{--++} = \mathbb{P}_{-+-+} = \mathbb{P}_{-++-} = p$$

and for the six remaining sign vectors

$$\mathbb{P}_{+++-} = \mathbb{P}_{++-+} = \mathbb{P}_{+-++} = \mathbb{P}_{---+} = \mathbb{P}_{--+-} = \mathbb{P}_{-+--}$$

Furthermore, by adding complementary challenges, we have that $p = p + 0 = \mathbb{P}_{+--+} + \mathbb{P}_{+---} = \mathbb{P}_{+--\cdot} = \frac{1}{8} - 3\frac{\arcsin\frac{1}{3}}{4\pi}$ using the generic formula for tri-variate Gaussian distributions. Therefore,

$$H_{C_4} = H_3 = -\left(1 - 6\frac{\arcsin\frac{1}{3}}{\pi}\right)\log\left(\frac{1}{8} - 3\frac{\arcsin\frac{1}{3}}{4\pi}\right) - 6\left(\frac{\arcsin\frac{1}{3}}{\pi}\right)\log\left(\frac{\arcsin\frac{1}{3}}{\pi}\right)$$

B. Case $n = 4$

Similar techniques have been employed in order to compute entropies with $n = 4$. Because $\frac{\arcsin(\frac{1}{2})}{\pi}$ is a rational number (it is in fact equal to $\frac{1}{6}$), the results for $n = 4$ are much simpler, compared to the case $n = 3$. We obtain the following results for the entropy while adding up sign vectors from 1 to 2^{n-1}:

Additional sign vector	entropy (bits)
$(1\,1\,1\,1)$	1
$(1\,-\,1\,-)$	2
$(1\,1\,-\,-)$	3
$(1\,-\,-\,1)$	4
$(1\,1\,1\,-)$	$\frac{43}{8} + \frac{\log 3}{2} - \frac{11}{24}\log 11$
$(1\,1\,-\,1)$	$\frac{16}{3} + \frac{1}{2}\log 3 - \frac{5}{12}\log 5$
$(1\,-\,1\,1)$	$\frac{71}{12} - \frac{\log 3}{8}$
$(-\,1\,1\,1)$	$\frac{14}{3} + \log 3$

In summary for $n = 1, 2, 3, 4$:

n	1	2	3	4
H_n	1	2	3.6655...	6.251...

III. Zero Probabilities

A simple method to bound from above the entropy H_n is to count the number of sign vectors with non-zero probability. Indeed, the entropy will be smaller than or equal to the logarithm of this count (also called the *max-entropy*). We found that it is possible to characterize such sign vectors, or more precisely, their complementary set in $\{\pm 1\}^M$.

Lemma 1. *Let $b = \{b_i\}_{i\in[1;M]}$ a sign vector. Then $\mathbb{P}_b = 0$ iff there exists $\alpha = (\alpha_1, ..., \alpha_M) \in \mathbb{R}^M \backslash \{0\}^M$ such that $\mathrm{sgn}(\alpha_i) = b_i$ when $\alpha_i \neq 0$ and $\sum_{i=1}^M \alpha_i c_i = 0$. We call such a vector α an **annihilator** (for C).*

Finding an upper bound on the entropy thus amounts to counting the sign vectors that verify this property. In order to compute the number of sign vectors with non-zero probabilities, we used the following property that allows to restrict the set of sign vectors that potentially have a non-zero probability. It assumes that $C = C^n$ is the matrix containing the first 2^{n-1} challenges in lexicographical order.

Lemma 2. *Let $n > 1$ and b a sign vector of size 2^{n-1}. Let b_l (resp. b_r) be the left (resp. right) half of the signs of b. Then $\mathbb{P}_b = 0$ if $\mathbb{P}_{b_l} = 0$ or $\mathbb{P}_{b_r} = 0$ when considering b_l and b_r as sign vectors for the matrix C^{n-1}.*

Interestingly, the Gaussian nature of X is irrelevant to the specific problem of counting sign vectors with non-zero probability. To verify that a sign vector has non-zero probability, one can simulate any continuous distribution, for example a uniform distribution, to speed up computation time.

Results for $n = 1$ to 8

n	Non-zero probabilities	Proportion among challenges	max-entropy
1	2	1	1
2	4	1	2
3	14	0.875	3.8073
4	104	0.40625	6.7004
5	1882	0.0287	10.8780
6	94572	$2.202 \cdot 10^{-5}$	16.5291
7	15028134	$8.147 \cdot 10^{-13}$	23.8411
8	8378070864	$2.462 \cdot 10^{-29}$	32.9640

For $n > 8$ the combinatorial problem soon becomes intractable (complexity $\sim 2^{66}$ for $n = 9$).

IV. Equivalent Probability Classes

There is an inherent symmetry in this problem. Indeed, reordering the random variables $X_1, ..., X_n$ does not change the entropy, and neither does replacing X_i with $-X_i$ because the Gaussian distribution is symmetric. This allows us to find sign vectors with equal probabilities. For the rest of the section, we will suppose that $M = 2^{n-1}$ and choose as challenges the first 2^{n-1} challenges in lexicographical order, starting with the all 1 challenge vector, up to $c_{2^{n-1}} = (1, -1, -1, ..., -1)$.

Let $\sigma \in \mathfrak{S}_n$ be a permutation, we define $X_\sigma = (X_{\sigma(1)}, ..., X_{\sigma(n)})^T$. Firstly consider σ to be a transposition, $\sigma = (i \quad j)$ and suppose $i \neq 1, j \neq 1$. Because of the aforementioned considerations, we have that CX and CX_σ have the same distribution. Let C_σ be the matrix obtained from C by applying σ on the columns (here, by swapping columns i and j). By definition, we have that $CX_\sigma = C_\sigma X$. Now, because C contains all rows starting with 1, since $1 \notin \{i, j\}$, C_σ can also be obtained by permuting certain rows of C. Let π be that row permutation, and $b = (b_1, b_2, ..., b_{2^{n-1}})$ a sign vector. Then, because CX and $C_\sigma X$ have same distribution, b and b_π, where b_π is obtained from b by applying π to the coordinates, have same probability.

If $1 \in \{i, j\}$, i.e. $\sigma = (1 \quad j)$ this cannot be directly applied since the lines of C and C_σ are not the same anymore. However, we can notice that if we multiply all the columns of C_σ by the $j-th$ column and call the new matrix C'_σ, then indeed C'_σ is obtained from C by permuting the lines. Thus, if π is the corresponding permutation, b and $(c_{1,j} b_{\pi(1)}, c_{2,j} b_{\pi(2)}, ..., c_{2^{n-1},j} b_{\pi(2^{n-1})})$ have the same probability. Since every permutation can be expressed as a composition of transpositions, composing the aforementioned transformations allows to express any permutation σ.

For the sign changes, take $s = (1, \pm 1, ..., \pm 1)$ a vector of n signs, and consider the vector $X^s = (s_1 X_1,\ s_2 X_2,\ \cdots,\ s_n X_n)^T$. Since the Gaussian distribution is symmetric, we have that CX and CX^s have the same distribution. Furthermore, let's denote by C^s the matrix obtained from C where the column i is multiplied by s_i. By definition, $C^s X = CX^s$. Now, C^s can also be obtained from C by permuting some lines. If π is the corresponding permutation, b and b_π have the same probability. Now, for a more general vector $s = (\pm 1, \pm 1, ..., \pm 1)$, we can simply consider $-s$ and then look at the permutation induced by $\tilde{s} = (1, -s_2, -s_3, ..., -s_n)$.

Definition 1. *We say that two sign vectors b and b' are **equivalent** if $\mathbb{P}_b$ and $\mathbb{P}_{b'}$ are found to be equal using the transformations described above. Note that this indeed defines an equivalence relation on the sign vectors. The **equivalence class** of b is then the set of vectors equivalent to b.*

We were able to determine equivalence classes up to $n = 7$. For example, for $n = 5$, there are 7 equivalence classes, as described below:

Class size	Probability per vector	Sign vector in class
10	0.0145269	$+++++++++++++++++$
160	0.0006334	$-++++++++++++++++$
320	0.0007351	$--+++++++++++++++$
960	0.0002285	$---++++++++++++++$
80	0.0022002	$----+++++++++++++$
320	0.0002961	$---+-+++++++++++$
32	0.0008077	$---+-+++-+++++++$

Probabilities have been obtained via Monte Carlo simulations. In order to estimate H_n, n Gaussian variables are sampled and the corresponding sign vector is determined. Then we evaluate which equivalence class this sign vector belongs to, and count the number of occurrences per sign vector. The probability for each sign vector is then estimated as the number of occurrences of the equivalence class, divided by the number of simulations and the number of elements in that class. This allows us to estimate the entropy up to $n = 7$.

Results for $n = 1$ to 7

n	Equivalence classes	Estimated entropy
1	1	1
2	1	2
3	2	3.6655
4	3	6.251
5	7	9.97
6	21	15.24
7	135	21.9

V. Perspective: Compressed Sensing

Computing the exact number of non-zero probabilities seems to become intractable for $n \geq 9$. Indeed, given the number of non-zero probabilities for $n = 8$, about 2^{66} sign vectors need to be checked in order to filter out those with zero probability. Finding clusters of sign vectors with identical probably also becomes a daunting task for $n \geq 8$. Therefore, it might be interesting to consider other methods to determine the maximum PUF entropy.

One such way could be a method inspired by compressed sensing. Let S be a sparse $h \times M$ matrix with coefficients in $GF(2)$, and $h < M$. If we multiply the bitvector B with S (here, B is seen as a random variable taking values in $GF(2)^M$), we obtain a new bitvector with entropy at much h. However, this entropy might be higher than the entropy of h challenges. We can hope to obtain a tight lower bound of H_n in this fashion. However, we would still need much more than 2^{H_n} simulations in order to obtain a good lower bound for H_n, which might also quickly become untractable.

We performed a certain number of compressed sensing simulations. Figure 1 shows the results for $n = 8$, with h varying between 1 and 21, and various weights per line for S. For large enough weights, the entropy seems to remain close to h in this domain. Thus, the bitvector obtained via compressed sensing has almost full entropy.

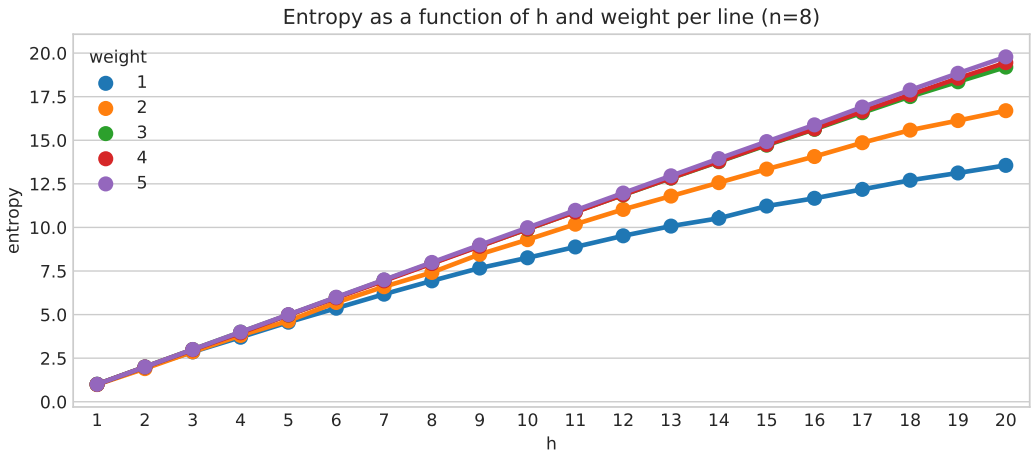

Fig. 1. Compressed sensing results for $n = 8$

VI. Conclusion

Despite the relatively simple formulation, the problem of computing the maximal entropy of all possible sign vectors generated by n Gaussian variables has very high a complexity, at the order of $2^{2^{n-1}}$. Thanks to a careful analysis of that problem, we were able to obtain exact expressions up to $n = 4$, very good approximations up to $n = 7$, and preliminary results for $n = 8$. However, an exact solution for larger values seems out of reach. Even determining the asymptotic behavior remains a difficult, open problem. The results we obtained so far seem to suggest a superlinear behavior for H_n, maybe even a quadratic growth, as shown in Figure 2. While the quadratic fit seems quite good for the results obtained so far, obtaining additional data points would increase our confidence in an asymptotically quadratic growth for H_n.

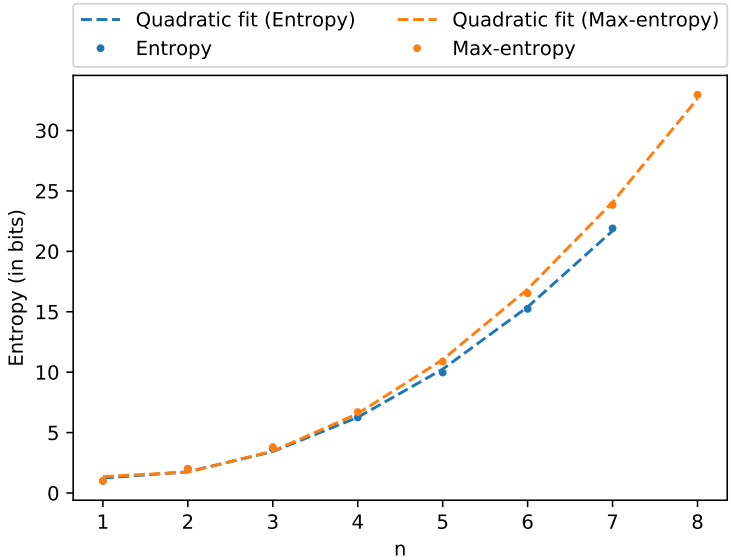

Fig. 2. Entropy and max-entropy, with the best-fit quadratic polynomial function.

References

[1] Z. Cherif, J.-L. Danger, S. Guilley, and L. Bossuet, "An easy-to-design PUF based on a single oscillator: the Loop PUF," in *15th Euromicro Conference on Digital System Design (DSD)*. IEEE, 2012, pp. 156–162.

[2] R. Maes and I. Verbauwhede, "Physically unclonable functions: A study on the state of the art and future research directions," in *Towards Hardware-Intrinsic Security*. Springer, 2010, pp. 3–37.

[3] O. Rioul, P. Solé, S. Guilley, and J.-L. Danger, "On the entropy of physically unclonable functions," in *IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2016, pp. 2928–2932.