Advances in Mathematics of Communications
Volume 14, No. 3, 2020, 491–505

doi:10.3934/amc.2020060

CHALLENGE CODES FOR PHYSICALLY UNCLONABLE FUNCTIONS WITH GAUSSIAN DELAYS: A MAXIMUM ENTROPY PROBLEM

Alexander Schaub*, Olivier Rioul and Jean-Luc Danger

LTCI, Telecom Paris, Institut Polytechnique de Paris
75013 Paris, France

Sylvain Guilley

Secure-IC S.A.S.
35510 Cesson-Sévigné, France

Joseph Boutros

Texas A&M University
23874 Doha, Qatar

Abstract. Motivated by a security application on physically unclonable functions, we evaluate the probability distributions and Rényi entropies of signs of scalar products of i.i.d. Gaussian random variables against binary codewords in $\{\pm 1\}^n$. The exact distributions are determined for small values of n and upper bounds are provided by linking this problem to the study of Boolean threshold functions. Finally, Monte-Carlo simulations are used to approximate entropies up to $n = 10$.

1. Introduction

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 $X_1, X_2, \dots, X_n \sim \mathcal{N}(0, 1)$. Let $X = (X_1, X_2, \dots, X_n)$ and consider the scalar products

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

and the associated sign bits

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

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

$$H(C) = H(B_1, B_2, \dots, B_M)? \tag{3}$$

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 largely combinatorial as shown below.

2020 *Mathematics Subject Classification:* Primary: 94A17, 94C10.

Key words and phrases: Multivariate Gaussian distribution, entropy, Boolean threshold functions.

* Corresponding author: Alexander Schaub.

1.1. Notations and definitions.

Definition 1.1 (Challenge code)**.** Let $n > 0, M > 0$ be two integers. A (n, M) *challenge code* C is a subset $C \subseteq \{-1, +1\}^n$ of cardinality M. The elements of this subset are called *codewords*, and the i-th codeword is denoted by c_i. By an abuse of notation, we identify the challenge code with the $n \times M$ matrix C, called the *challenge matrix*, which rows contain all codewords exactly once. The i-th row is c_i, and conversely, for any codeword $c \in C$, $i(c)$ denotes its row index.

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 [14]. More formally, a PUF is a function that takes several challenges $c_1, c_2, \dots, c_M$ (the so-called *challenge code*) as inputs and returns the bitvector identifier $(b_1, b_2, \dots, b_M)$ [19]. PUFs exploit small, uncontrollable physical variations of the manufacturing process that cannot be replicated, hence the name "physically unclonable".

Definition 1.2 (Physically unclonable function (PUF))**.** Let C be an (n, M) challenge code. Let $x = (x_1, \dots, x_n) \in \mathbb{R}^n$ such that the scalar product $c \cdot x \neq 0$ for all codewords $c \in C$. Then the *physically unclonable function* (PUF) with parameter x is the function $f^x : C \to \{-1, +1\}$ defined as

$$f^x(c) = \operatorname{sgn}(c \cdot x), \tag{4}$$

where sgn is the sign function and $\cdot$ denotes the usual scalar product. Equivalently, f^x is given by the *sign vector* $b = (b_1, b_2, \dots, b_M) \in \{-1, +1\}^M$ such that

$$b_i = \operatorname{sgn}(c_i \cdot x) \qquad (i = 1, \dots, M). \tag{5}$$

The following notion of *randomized* PUF coincides with that of a PUF at a design stage, when it is not yet instantiated by a foundry fabrication process (cf. [7, Fig. 1]).

Definition 1.3 (Randomized PUF)**.** For a fixed (n, M) challenge code, we define the *random PUF* as f^X, where $X = (X_1, X_2, \dots, X_n)$ and X_i are i.i.d. standard normal random variables $X_i \sim \mathcal{N}(0, 1)$.

The corresponding random sign vector is then $B = (B_1, \dots, B_M)$, where

$$B_i = \operatorname{sgn}(c_i \cdot X) \qquad (i = 1, \dots, M) \tag{6}$$

with probability distribution

$$\mathbb{P}_b = \mathbb{P}[B = b] = \mathbb{P}[B_1 = b_1, B_2 = b_2, \dots, B_M = b_M] \qquad (b \in \{-1, +1\}^M).$$

We denote $|\mathrm{supp}(\mathbb{P}_b)|$ the cardinality of the support of $\mathbb{P}_b$.

To assess the security of a PUF, it is necessary that the entropy of the identifier's distribution is sufficiently high. The most natural definition is the Shannon entropy, characterizing the uncertainty about the PUF distribution. Depending on the desired application, other kinds of entropies may be relevant. The most conservative view is to consider the *min-entropy* H_∞, which can be interpreted as the "cloning" entropy in the *worst* case, when the PUF to clone is obtained with probability $\max_{b \in \{\pm 1\}} \mathbb{P}_b$. When using a PUF to generate a key, the min-entropy also characterizes the security of the key, as shown for example in [9]. In other settings, the *collision entropy* allows for a more accurate security bound on the key derivation, as suggested by Skorski [20] and Dodis *et al.* [10]. It accounts for PUF uniqueness, since it is related to the probability that no two generated keys are the same. In contrast, the max-entropy H_0 has no obvious practical interest apart from being an easily computable upper-bound of the Shannon entropy (and of all other Rényi entropies).

Definitions for the different kinds aforementioned entropies are given below. Each depends on the choice of a challenge code C.

Definition 1.4 (Rényi entropies [17]). For $\alpha \geq 0$, the *Rényi entropy* of order α is defined as

$$H_\alpha(C) = \frac{1}{1-\alpha} \log_2 \sum_{b \in \{\pm 1\}^M} \mathbb{P}_b^\alpha .$$

As special cases (taking the limits when α approaches 1 or infinity) we have

$$\begin{aligned}
H_0(C) &= \log_2 |\mathrm{supp}(\mathbb{P}_b)| && \text{(max-entropy)}\\
H_1(C) = H(C) &= \sum_{b \in \{\pm 1\}^M} \mathbb{P}_b \log_2 \frac{1}{\mathbb{P}_b} && \text{(Shannon entropy)}\\
H_2(C) &= -\log_2 \sum_{b \in \{\pm 1\}^M} \mathbb{P}_b^2 && \text{(collision entropy)}\\
H_\infty(C) &= -\log_2 \max_{b \in \{\pm 1\}^M} \mathbb{P}_b && \text{(min-entropy).}
\end{aligned}$$

A well-known property of the Rényi entropies is that H_α is non-increasing in α. Thus, for any code C, $H_\infty(C) \leq H_2(C) \leq H(C) \leq H_0(C)$. It is also easily seen that $H_2(C) \leq 2H_\infty(C)$.

Definition 1.5 (Full entropy). For any $\alpha \geq 0$, we define the *full entropy* $H_\alpha(n)$ as the Rényi entropy for the $(n, 2^n)$ challenge code that contains all possible codewords.

The *full entropy* is highest among all codes, as shown in the following lemma.

Lemma 1.6 (Full entropy is maximal). *For any $\alpha \geq 0$ and any challenge code C,*

$$H_\alpha(n) \geq H_\alpha(C).$$

Proof. We prove a stronger result: For any challenge matrix C of an (n, M) challenge code and challenge matrix C' of an $(n, M+1)$ challenge code where the first M lines are identical to C, $H_\alpha(C') \geq H_\alpha(C)$.

Let b be a sign vector associated with C such that $\mathbb{P}_b > 0$, and let b^+ (resp. b^-) the sign vector associated with C' equal to $(b_1, \dots, b_M, 1)$ (resp. $(b_1, \dots, b_M, -1)$). By definition of C', one has $\mathbb{P}_b = \mathbb{P}_{b^+} + \mathbb{P}_{b^-}$.

Assume $\alpha > 1$. To prove that $\mathbb{P}_b^\alpha \geq \mathbb{P}_{b^+}^\alpha + \mathbb{P}_{b^-}^\alpha$, consider $\frac{\mathbb{P}_{b+}}{\mathbb{P}_b}$ and $\frac{\mathbb{P}_{b-}}{\mathbb{P}_b}$. Since $0 \leq \frac{\mathbb{P}_{b+}}{\mathbb{P}_b}, \frac{\mathbb{P}_{b-}}{\mathbb{P}_b} \leq 1$, we know that $(\frac{\mathbb{P}_{b+}}{\mathbb{P}_b})^\alpha \leq \frac{\mathbb{P}_{b+}}{\mathbb{P}_b}$ and $(\frac{\mathbb{P}_{b+}}{\mathbb{P}_b})^\alpha \leq \frac{\mathbb{P}_{b+}}{\mathbb{P}_b}$. Therefore,

$$(\frac{\mathbb{P}_{b+}}{\mathbb{P}_b})^\alpha + (\frac{\mathbb{P}_{b-}}{\mathbb{P}_b})^\alpha \leq \frac{\mathbb{P}_{b+}}{\mathbb{P}_b} + \frac{\mathbb{P}_{b-}}{\mathbb{P}_b} = 1. \tag{7}$$

which implies $\mathbb{P}_{b+}^\alpha + \mathbb{P}_{b-}^\alpha \leq \mathbb{P}_b^\alpha$. Summing over all $\mathbb{P}_b$ we obtain

$$\sum_{b \in \{\pm 1\}^M} \mathbb{P}_b^\alpha = \sum_{b \in \{\pm 1\}^M} (\mathbb{P}_{b^+} + \mathbb{P}_{b^-})^\alpha \geq \sum_{b \in \{\pm 1\}^M} \mathbb{P}_{b^+}^\alpha + \mathbb{P}_{b^-}^\alpha \geq \sum_{b \in \{\pm 1\}^{M+1}} \mathbb{P}_b^\alpha . \tag{8}$$

The assertion follows by taking the logarithm on both sides of this inequality and multiplying by the negative constant $\frac{1}{1-\alpha}$.

The case $\alpha < 1$ is similar: The inequalities (7) and (8) are reversed because $x^\alpha \geq x$ for $x \in [0,1]$, but the constant $\frac{1}{1-\alpha}$ is positive. Therefore, the same assertion follows. The cases $\alpha = 1$ and $\alpha = \infty$ are established by taking limits. □

Notice that the maximum entropy $H^\alpha(n)$ is always attained by a $(n, 2^{n-1})$ challenge code, by the following symmetry argument: since $\operatorname{sgn}(c \cdot x) = -\operatorname{sgn}((-c) \cdot x)$, the set $\{\pm 1\}^n$ can be partitioned into two opposite sets where codewords in the second set bring no additional entropy. Indeed, adding a codeword c to a code C which already contains $-c$ does not change the probabilities of the sign vectors, only their labeling. This leaves all Rényi entropies unchanged. Therefore, it is possible to obtain the maximum entropy with any $(n, 2^{n-1})$ code $\mathcal{C}$ satisfying $c \in \mathcal{C} \implies -c \notin \mathcal{C}$.

Table 1 summarizes the notations used in the remainder of this paper.

Table 1. Summary of Notations.

Notation	Explanation
n	number of delay elements in a PUF
X_i	Gaussian random variable representing the delay difference of the i-th delay element ($i \in [1, n]$)
X	$X = (X_1, X_2, \ldots, X_n)$
x_i	realization of X_i
M	number of challenges
C	challenge code, a matrix defined by its rows $(c_i)_{i \in [1,M]}$
sgn	$\operatorname{sgn}(x) = 1$ if $x > 0$, $\operatorname{sgn}(x) = -1$ if $x < 0$, and $\operatorname{sgn}(0) = 0$.
B_i	$B_i = \operatorname{sgn}(c_i \cdot X)$
B	$B = (B_1, B_2, \ldots, B_M)$
b_i	realization of B_i
b	realization of B
$\mathbb{P}_b$	$\mathbb{P}_b = \mathbb{P}[B = b]$

1.2. Motivation.

Definitions 1.2 and 1.3 correspond to a particular PUF that exploits the variability of n distinct delay elements (a so-called "Loop PUF"), where $X_1, X_2, \ldots, X_n$ are independent *Gaussian delay* differences. This type of PUF has been first described by Cherif *et al.* [5]. A previous modelization of the Loop PUF, obtained via Monte-Carlo simulations of the possible circuit behaviors, showed a distribution of delays close to a Gaussian distribution, as shown in Figure 1. Other types of simulations also suggest a Gaussian distribution of process variations, and thus delay differences, in electronic circuits [4]. This motivates the choice of modeling the delay differences of the Loop-PUF as independent Gaussian variables.

Because they share the mathematical model with the Loop-PUF, definitions 1.2 and 1.3 also apply to the Arbiter PUF [11], for which the Gaussian model has been confirmed [15, 22], and to the RO-sum PUF [25].

These process variations can then be exploited in different ways. For example, it is possible to build authentication protocols based on PUFs: an authentication server queries a PUF via a set of challenges and checks the PUF answer against a whitelist. In this way, counterfeit or overproduced chips can be detected. This requires no implementation of costly asymmetric cryptography primitives, and is therefore adapted to low-cost IoT devices. The PUF can also be used to generate a

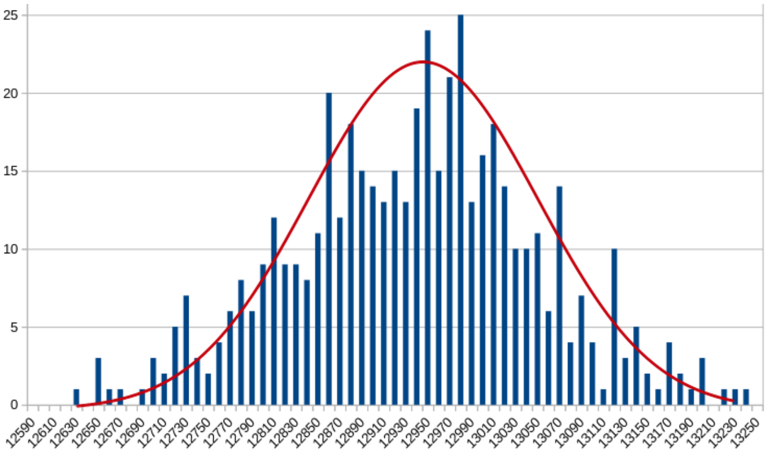

Figure 1. Distribution of delays obtained via circuit simulation

secret cryptographic key that is required for secure storage or communications with other devices. Using a PUF is more secure than directly storing the cryptographic key into memory, from where it might potentially be read or written by an attacker.

1.3. State of the art.

1.3.1. *Results on the min-entropy.* An upper bound of the min-entropy has been derived for the so-called RO-sum PUF by Delvaux *et al.* in [8]. Since this PUF shares the same mathematical bound as the Loop-PUF, this result is also relevant for our analysis. The following upper-bound is valid for odd values of n:

$$H_\infty(n) \le -\log_2\left(\frac{1}{2}\left(1-\sqrt{\frac{n-1}{n}}\sum_{i=0}^{(n-3)/2}\frac{(2i)!}{(i!)^2(4n)^i}\right)\right). \tag{9}$$

This expression is not easy to interpret, but we have the following bound for practical values of n:

$$H_\infty(n) \le 4n \quad \text{for } n \le 251. \tag{10}$$

The min-entropy is therefore at most linear in n for $n \le 251$. Because of the inequality $H_2 \le 2H_\infty$, valid for any distribution, we deduce the following bound on the collision entropy:

$$H_2(n) \le 8n \quad \text{for } n \le 251. \tag{11}$$

1.3.2. *Exact values for small n.* Exact results for the entropy and probability distribution of the Loop-PUF have been obtained in certain special cases. Rioul *et al.* showed in [18] that the optimal challenge code when $M \le n$ is given by a Hadamard code[1] C for which one can attain a uniform distribution of the Loop-PUFs, giving

$$H(C) = n.$$

The exact calculation of the PUF distribution of n delay elements for $M \ge n$ can be carried out only for very small values of n. Rioul *et al.* [18] give the exact

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

values of the Loop-PUF distribution, and thus $H(C)$ for all $n, M \leq 3$ using well-known closed-form formulas for orthant probabilities of bi- and trivariate normal distributions (see Lemma 2.1).

1.3.3. *Results on the max-entropy.* The max-entropy $H_0(n)$ is simply the logarithm of the number of different Loop-PUFs of n delay elements. This number has been computed for small values of $n \leq 10$, because it actually corresponds to the number of so-called Boolean Threshold Functions (BTF) of $n-1$ variables. This number was determined up to $n = 8$ by Winder [24], up to $n = 9$ by Muroga *et al.* [16] and finally up to $n = 10$ by Gruzling [12]. Asymptotic estimates have also been published [26]. These results are recalled in Section 3.

Unfortunately, the quadratic behavior in n of the max-entropy $H_0(n)$ somehow overestimates the security of the PUFs, since it is much higher than the min-entropy, which is approximatively linear in n.

1.4. OUR CONTRIBUTIONS. In this work, we extend previous results in two directions.

First, we provide the exact values of the distribution of the Loop-PUF (for all possible challenges) for $n = 3$ and $n = 4$. This allows us to compute the exact values of all entropies in these cases. Such an exact computation comes as a surprise since no closed-form expression exists for the orthant probabilities of an M-dimensional Gaussian vector for $M \geq 4$. In our computation, we leverage on the discrete nature of the challenge code to determine these probabilities up to $M = 8$.

Second, we introduce a novel algorithm for the simulation of equivalence classes (SEC). The SEC algorithm also finds all equivalence classes of challenge codewords corresponding to the same value of joint probabilities $\mathbb{P}_b$. Interestingly, this problem is purely of discrete combinatorial nature. The actual values of the corresponding probabilities are then estimated by Monte Carlo simulation, which allows us to compute all relevant entropies. We provide the resulting values of the entropies $H_0(n)$, $H(n)$, and $H_2(n)$ up to $n = 10$.

The remainder of the paper is organized as follows. Section 2 presents exact values of the distributions and entropies for the cases $n = 3$ and $n = 4$. Section 3 recalls results obtained from the study of Boolean threshold functions which will be used later on. The SEC algorithm is presented in Section 4 along with the entropies up to $n = 10$. Section 5 concludes.

2. CLOSED-FORM EXPRESSIONS

2.1. PRELIMINARIES. In order to determine the closed-form expressions of the PUF distributions up to $n = 4$, we need the following lemmas.

Lemma 2.1 (Orthant probabilities for the bi- and trivariate normal distribution)**.** *Let $n > 0$, c_1 and c_2 two challenges, and $Y_1 = c_1 \cdot X, Y_2 = c_2 \cdot X$. Let $\rho = \frac{\mathbb{E}[Y_1 Y_2]}{n}$ the correlation coefficient of Y_1 and Y_2. Then*

$$\mathbb{P}[Y_1 > 0, Y_2 > 0] = \mathbb{P}_{++} = \frac{1}{4} + \frac{\arcsin(\rho)}{2\pi}. \tag{12}$$

Let c_3 be a third challenge vector and $Y_3 = c_3 \cdot X$, and denote the correlation coefficients between Y_i and Y_j by $\rho_{i,j} = \frac{\mathbb{E}[Y_i Y_j]}{n}$. Then

$$\begin{aligned} \mathbb{P}[Y_1 > 0, Y_2 > 0, Y_3 > 0] &= \mathbb{P}_{+++} \\ &= \frac{1}{8} + \frac{\arcsin(\rho_{1,3}) + \arcsin(\rho_{1,2}) + \arcsin(\rho_{2,3})}{4\pi}. \end{aligned} \tag{13}$$

The bivariate case was already known to Hermite [21]. The extension to the trivariate case is a lesser known extension and can be found, for instance, in [3]. A short proof of both formulas is given by Rioul *et al.* in [18].

Lemma 2.2 (Zero probabilities)**.** *Let* $b = (b_i)_{i\in[1;M]}$ *be a sign vector. Then* $\mathbb{P}_b = 0$ *if and only if there exists* $\alpha = (\alpha_1, \dots, \alpha_M) \in \mathbb{R}^M \backslash \{0\}^M$ *such that* $\operatorname{sgn}(\alpha_i) = b_i$ *when* $\alpha_i \neq 0$ *and* $\sum_{i=1}^M \alpha_i c_i = 0$*.*

Proof. Suppose that such a vector α exists. There is at least one component that is different from 0. Without loss of generality, suppose that $\alpha_1 \neq 0$. We then have

$$c_1 = -\frac{1}{\alpha_1}\sum_{i=2}^M \alpha_i c_i.$$

In particular, this implies that

$$X \cdot c_1 = -\frac{1}{\alpha_1}\sum_{i=2}^M \alpha_i (c_i \cdot X).$$

Now, if $\forall i > 1,$ such that $\alpha_i \neq 0, \operatorname{sgn}(\alpha_i) = \operatorname{sgn}(c_i \cdot X)$, the sign of the right-hand side of the expression is the opposite sign of α_1. Thus, $\alpha_1 = -\operatorname{sgn}(c_1 \cdot X)$, which contradicts our hypothesis.

Conversely, suppose that $\mathbb{P}_b = 0$. Therefore, the Gaussian vector $(c_i \cdot X)_i$ is degenerate, and its support is included in a sub-space of $\mathbb{R}^M$ of dimension $< M$. In particular, it is included in some hyperplane of equation $\sum_{i=1}^M a_i x_i = 0$, where the a_i are not all 0. Since $\mathbb{P}_b = 0$, this hyperplane is disjoint from the orthant defined by the signs of b, that is the set $x_1 b_1 > 0, x_2 b_2 > 0, \dots, x_M b_M > 0$. Therefore, we have that all $a_i b_i$ have the same sign, that we can take positive. Since the support is included in the hyperplane defined before, we must have

$$\sum_{i=1}^M a_i (c_i \cdot X) = (\sum_{i=1}^M a_i c_i) \cdot X = 0$$

for all $X \in \mathbb{R}^n$, and therefore $\sum_{i=1}^M a_i c_i = 0$. By setting $\alpha_i = a_i$, the α_i have the same signs as the b_i and are not all 0, which concludes the proof. □

Lemma 2.3 (Equivalence classes)**.** *Suppose that after permuting and/or changing the signs of certain* columns *of* C*, one obtains a matrix* C' *that can be obtained by permuting, and then optionally changing the signs, of certain* lines *from* C*. Denote the corresponding permutation of the lines by* $\sigma \in S_M$*, and the following change of signs of the lines by* $s_i \in \{\pm 1\}^M$*. Then for any sign vector* $b = (b_i)_{i\in[1;M]}$*,* b *has the same probability as*

$$b' = (s_1 b_{\sigma(1)}, s_2 b_{\sigma(2)}, \dots, s_M b_{\sigma(M)}).$$

Such b *and* b' *are then said to be in the same equivalence class, or simply equivalent.*

Proof. Permuting the columns or changing corresponds to a permutation or sign changes of the X_i. For any $s = (s_1, s_2, \dots, s_n) \in \{-1, +1\}^n$ and $\sigma \in S_n$, the joint distribution of $X = (X_i)_i$ and $(s_i X_{\sigma(i)})$ are the same. □

2.2. 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 trivariate Gaussian, recalled in Equation (13). This yields an entropy of

$$H(C_3) = -\Big(\frac{1}{4} - 3\frac{\arcsin\frac{1}{3}}{2\pi}\Big)\log\Big(\frac{1}{8} - 3\frac{\arcsin\frac{1}{3}}{4\pi}\Big). \tag{14}$$

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

$$\mathbb{P}_{+---} = \mathbb{P}_{-+++} = 0.$$

By exploiting symmetries, it follows that eight sign vectors satisfy

$$\begin{aligned} &\mathbb{P}_{++++} = \mathbb{P}_{++--} = \mathbb{P}_{+-+-} = \mathbb{P}_{+--+} \\ &= \mathbb{P}_{----} = \mathbb{P}_{--++} = \mathbb{P}_{-+-+} = \mathbb{P}_{-++-} = p \end{aligned}$$

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 trivariate normal distributions.

These findings are summarized in the table below.

Table 2. Distribution for $n = 3$

Size of equivalence class	Probability per element
8	$\frac{1}{8} - 3\frac{\arcsin\frac{1}{3}}{4\pi}$
6	$\frac{\arcsin\frac{1}{3}}{\pi}$

Therefore,

$$\begin{aligned} &H(C_4) = H(3) = \\ &-\left(1 - 6\frac{\arcsin\frac{1}{3}}{\pi}\right)\log\left(\frac{1}{8} - 3\frac{\arcsin\frac{1}{3}}{4\pi}\right) - 6\left(\frac{\arcsin\frac{1}{3}}{\pi}\right)\log\left(\frac{\arcsin\frac{1}{3}}{\pi}\right). \end{aligned} \tag{15}$$

2.3. 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 $n = 3$.

In order to compute the distribution for the maximal challenge code, we first determine the distributions of smaller codes. Sign vectors with zero probability are determined according to Lemma 2.2. Those of equal probability are found with the help of Lemma 2.3. Using recurrence relations between the probabilities, we are then able to deduce the sign vector distribution for larger codes, when adding one codeword each time.

The first four codewords that are chosen are the lines of a Hadamard matrix of order 4:

$$C_4 = \left(\begin{smallmatrix} 1 & 1 & 1 & 1 \\ 1 & - & 1 & - \\ 1 & 1 & - & - \\ 1 & - & - & 1 \end{smallmatrix}\right).$$

As recalled before, the sign vector distribution is uniform for this challenge matrix.

The results when adding one additional codeword are summarized below. For the sign vectors, it is understood that the opposite sign vectors are also present in each probability class.

- Additional codeword $(1\ 1\ 1\ -)$:

Probability per sign vector	Sign vectors
$\frac{11}{192}$	+++++,++−−+,+−+−+,+−−+−
$\frac{1}{192}$	++++−,++−−−,+−+−−,+−−++
$\frac{1}{32}$	++−++,++−+−,+−+++,+−++−,+−−−+,+−−−−
$\frac{1}{16}$	+++−+

- Additional codeword $(1\ 1\ -\ 1)$:

Probability per sign vector	Sign vectors
$\frac{10}{192}$	++++++,++−−+−,+−+−++,+−−+−+
$\frac{1}{192}$	+++++−,++++−+,++−−++,++−−−−, +−+−+−,+−+−−+,+−−+++,+−−+−−
$\frac{1}{32}$	+++−++,+++−+−,+−++++,+−++−+
$\frac{1}{64}$	++−+++,++−++−,++−+−+,++−+−−, +−−−++,+−−−+−,+−−−−+,+−−−−−

- Additional codeword $(1\ -\ 1\ 1)$:

Probability per sign vector	Sign vectors
$\frac{3}{64}$	+++++++,++−−+−+,+−−+−++,+−+−++−
$\frac{1}{192}$	+++++−+,+−+−+−−,+−−++++,+−+−+++, +−−+−−+,++−−−−+,++++++−,++++−++, +−−−−−−,++−−+−−,+−−−+−+,+−−+−+−, ++−−+++,+−−−++−,+−+−−+−,+−−−−++
$\frac{1}{96}$	+−−−+++,+−−−−−+,+−−−−+−,+−−−+−−
$\frac{1}{64}$	++−+−++,+−++−++,+++−+−+,+++−++−, ++−++−+,+−++++−,+−++−+−,++−+−−+, +++−+++,+−+++++,+++−+−−,++−++++

- Additional codeword $(-\ 1\ 1\ 1)$: This code maximizes the entropies. As there are too many sign vectors per equivalence class to enumerate them all, we give here only their numbers.

Probability per sign vector	Number of sign vectors
$\frac{1}{24}$	8
$\frac{1}{192}$	64
$\frac{1}{96}$	32

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

TABLE 3. Exact entropies for $n \leq 4$

n	1	2	3	4
$H(n)$	1	2	3.6655...	6.2516...
$H_0(n)$	1	2	3.8073...	6.7004...
$H_2(n)$	1	2	3.54615...	5.71049...
$H_\infty(n)$	1	2	3.20858...	4.58496...

3. RESULTS FROM THE THEORY OF BOOLEAN THRESHOLD FUNCTIONS

Boolean threshold functions (BTF) are a special class of Boolean functions that have been studied at least since the early 1950's. They have a special significance in several domains, such as building Boolean circuits [23], but also in the domain of machine learning [6]. More recently, they have even been studied in game theory [13]. There are several equivalent definitions of BTF. We adopt the following one.

Definition 3.1 (Boolean Threshold Function)**.** Let $n > 0$. A Boolean function $g : \{-1, +1\}^{n-1} \to \{-1, +1\}$ is said to be a *Boolean threshold function* of $n - 1$ variables if there exists a vector of $n - 1$ real numbers, $w = (w_1, \ldots, w_{n-1})$, called the *weights* of the BTF, as well as a real number w_0, called the *threshold*, such that:

$$\forall c \in \{-1, +1\}^{n-1}, g(c) = \begin{cases} 1 \text{ when } c \cdot w > w_0 \\ -1 \text{ when } c \cdot w < w_0 \end{cases} .$$

We have the following equivalence between BTFs with $n-1$ variables and PUFs with n elements:

Proposition 1. *Let C be the $(n, 2^{n-1})$ challenge code containing all codewords starting with 1. Then for any sign vector b, $\mathbb{P}_b > 0$ if and only if there exists a BTF of $n - 1$ variables represented by b, that is, the BTF g such that*

$$g(c') = b_{i(1,c'_1,c'_2,\ldots,c'_{n-1})} \qquad (\forall c' \in \{\pm 1\}^{n-1})$$

where $i(c)$ represents the index of the codeword $c = (1, c')$ in the challenge matrix C (see Definition 1.1).

Proof. First, suppose that $\mathbb{P}_b > 0$. Thus, there exists $x \in \mathbb{R}^n$ such that $\text{sgn}(c \cdot x) = b_{i(c)}$ for all $c \in C$. Let $w_0 = -x_1$ and for all $i \in [1, n-1], w_i = x_{i+1}$ and $c'_i = c_{i+1}$. Then

$$c \cdot x > 0 \iff 1 \cdot x_1 + \sum_{i=2}^{n} c_i \cdot x_i > 0 \iff \sum_{i=2}^{n} c_i \cdot x_i > -x_1$$

$$\iff \sum_{i=1}^{n-1} c'_i \cdot w_i > w_0 \iff c' \cdot w > w_0.$$

Thus, the BTF g, defined by the weights $(w_1, \ldots, w_n)$ and threshold w_0, is such that $g(c') = 1$ exactly where $b_{i(c)} = 1$.

Conversely, if there is a BTF corresponding to b, as shown above, there is at least one element $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ such that $b_{i(c)} = \text{sgn}(c \cdot x)$ for all $c \in C$. Let

$E_b = \{x \in \mathbb{R}^n : b_{i(c)} = \mathrm{sgn}(c \cdot x),\, c \in C\}$. Then, by hypothesis, E_b is not empty. It is also an open set, as the preimage of the open set $\mathbb{R}^{+*}$ by the continuous function $x \in \mathbb{R}^n \mapsto b_{i(c)}\,(c \cdot x)$. As a non-empty open set, E_b has non-zero volume. Furthermore, since the multivariate Gaussian distribution is non-degenerate, it has a non-zero probability on E_b. Thus, $\mathbb{P}_b > 0$. □

Two results from the analysis of BTF are relevant for the study of Loop-PUFs. First, the exact value for the number of BTF of $n-1$ variables has been computed up to $n = 10$ [12]. This gives the exact max-entropy for the Loop-PUF up to $n = 10$, and thus also an upper bound for the other entropies (Shannon, collision entropy, min-entropy). Results are shown below.

Table 4. Non-zero probabilities for $n = 1$ to 10

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	15 028 134	$8.147 \cdot 10^{-13}$	23.8411
8	8 378 070 864	$2.462 \cdot 10^{-29}$	32.9640
9	17561539552946	$1.517 \cdot 10^{-64}$	43.997
10	144130531453121108	$1.075 \cdot 10^{-137}$	57.000

The number of non-zero probabilities is referenced on Sloane's On-line Encyclopedia of Integer Sequences (OEIS) as sequence A000609 [1].

Second, asymptotic expressions have also been derived [26]:

$$\lim_{n\to\infty} \frac{H_0(n)}{n^2} = 1. \tag{16}$$

Therefore, the max-entropy is close to n^2 for large values of n. However, the min-entropy is only linear in n [8]. Because of this gap in the different entropies, a more careful analysis is necessary in order to determine exact values and estimates of the Shannon and collision entropies.

4. Equivalent probability classes

There is an inherent symmetry in the PUF 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 $\sigma \in S_n$ 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 some 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 $1 \in \{i, j\}$, 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 jth 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 $X^s = (s_1X_1,\ s_2X_2,\ \cdots,\ s_nX_n)^T$. Since the Gaussian distribution is symmetric, we have that CX and CX^s have the same distribution. Furthermore, denote by C^s the matrix obtained from C where the column i is multiplied by s_i. By definition, $C^sX = CX^s$. 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. For a vector s of the form $(-1, \pm 1, ..., \pm 1)$, we can simply look at the permutation induced by $-s = (1, -s_2, -s_3, ..., -s_n)$.

Definition 4.1. We say that two sign vectors b and b' are **equivalent** if b can be obtained from b' by the actions of the permutations σ and sign changes s. This defines an equivalence relation on the sign vectors. All sign vectors of a same equivalence class have same probability.

We were able to determine equivalence classes up to $n = 10$. 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	$---+-+++-+++++++$

Our simulation of equivalence classes (SEC) algorithm used to evaluate the Loop-PUF distribution consists in two steps.

1. During the first step, n independent standard normal variables are repetitively sampled. We then take their absolute values, sort them, and record the corresponding sign vector. Because changing the signs and re-ordering the X_i does not change the equivalence class, the two sign vectors corresponding to the X_i before and after these transformations are equivalent. This way, the same sign vector is always recorded for each equivalence class. This first step therefore allows us to estimate the probabilities of all equivalence classes.
2. Second, the algorithm determines the size of each equivalence class. This is necessary in order to estimate the probabilities of individual sign vectors. We use the same method as employed to evaluate the number of Boolean threshold functions, which is described, for instance, by Gruzling [12], section 3.1.2. The

estimated probability of a sign vector is then simply the number of occurrences of the equivalence class, divided by the total number of simulations and by the number of elements in that class.

This allows us to estimate the entropy up to $n = 10$. Note that the number of equivalence classes corresponds to the sequence A001532 on Sloane's OIES [2].

Table 5. Estimated Entropies for $n = 1$ to $n = 10$.

n	Equivalence classes	Shannon entropy	Collision entropy
1	1	1	1
2	1	2	2
3	2	3.665	3.545
4	3	6.250	5.708
5	7	10.015	8.456
6	21	15.189	11.600
7	135	21.956	14.890
8	2470	30.564	18.548
9	175428	41.038	22.231
10	52980624	53.47	26.06

All results obtained so far are summarized in Figure 2.

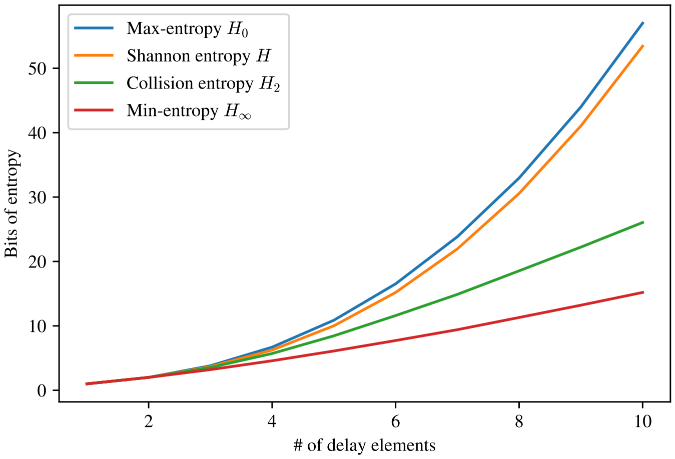

Figure 2. Comparison of entropies up to $n = 10$.

For cryptographic applications, a key should typically have at least 80 bits of entropy. Therefore, a PUF with $n = 10$ is insufficient. However, given our findings, a PUF with $n = 12$ or $n = 13$ is very likely to exceed this value, which is very interesting from an implementation complexity perspective.

5. Conclusions and perspectives

The exact values for the probabilities of all the sign vectors were determined for n up to 4. The methods employed might be applied for larger values of n. It is not known, however, if enough equations can be obtained this way to compute the probabilities of all sign vectors. The success of this method might also depend on the order in which challenges are added to the challenge code.

While a naive method would have complexity $O(2^{2^{n-1}})$, the SEC algorithm allows to estimate entropies reliably up to $n = 10$. For larger values of n, however, the SEC algorithm might not be feasible. Using (16) and a quadratic fit on the logarithm of the number of BTF, we can estimate the number of non-zero probabilities for $n = 11$ to be about 2^{77}. The size of each equivalence class does not exceed $2^n n!$, the number of pairs of permutations and sign changes. There are thus at least $1.8 \cdot 10^{12}$ equivalence classes for $n = 11$. Estimating their probabilities individually becomes intractable in time but also in space. Asymptotic formulas for the entropy and collision entropy are therefore necessary to assess the security of the Loop-PUF for larger values of n.

As a perspective, determining the entropies of the Loop-PUF when considering smaller challenge matrices would be of practical interest. Indeed, using less challenges would decrease the time necessary, for instance, to generate a cryptographic key from the PUF. The question of how many challenges to choose, and which ones maximize the entropies, should be addressed in future research. One such solution is a greedy approach, experienced by Rioul *et al.* in [18]. This leads to a piecewise-Hadamard matrix for the challenge matrix, and an almost linear increase in entropy when considering less than $2n$ codewords.

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 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 $n = 4$, and tight approximations up to $n = 10$. However, an exact solution for larger values seems out of reach. Even determining the asymptotic behavior remains an open problem. While it is known that the max-entropy is quadratic in n, and the min-entropy approximately linear in n, asymptotic expressions for the Shannon and collision entropy have not been determined yet. In particular, the Shannon entropy seems to be quadratic in n, which is a very good result for chip designers, since it would allow the production of high security circuits while keeping the number of elements per circuit small.

Acknowledgments

The authors would like to thank Gadiel Seroussi, who first suggested a possible link between the Loop-PUF and Boolean threshold functions at the LAWCI'18 conference in Campinas, Brazil.

References

[1] The On-Line Encyclopedia of Integer Sequences. A000609.

[2] The On-Line Encyclopedia of Integer Sequences. A001532.

[3] I. G. Abrahamson, Orthant probabilities for the quadrivariate normal distribution, *The Annals of Mathematical Statistics*, **35** (1964), 1685–1703.

[4] H. L. Chang and S. S. Sapatnekar, Statistical timing analysis considering spatial correlations using a single pert-like traversal, *Proceedings of the 2003 IEEE/ACM International Conference on Computer-aided Design, ICCAD '03,Washington, DC, USA, IEEE Computer Society*, (2003), 621–625.

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

[6] T. M Cover, Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition, *IEEE Transactions on Electronic Computers*, **14** (1965), 326–334.

[7] J.-L. Danger, S. Guilley, P. Nguyen and O. Rioul, PUFs: Standardization and evaluation, *Proc. 2nd IEEE Workshop on Mobile System Technologies (MST 2016), Milano, Italy*, (2016), 12–18, http://perso.telecom-paristech.fr/~rioul/publis/201609dangerguilleynguyenrioul.pdf, http://dx.doi.org/10.1109/MST.2016.11.

[8] J. Delvaux, D. Gu and I. Verbauwhede, Upper bounds on the min-entropy of RO Sum, Arbiter, Feed-Forward Arbiter, and S-ArbRO PUFs, *Hardware-Oriented Security and Trust (AsianHOST), IEEE Asian*, (2016), 1–6.

[9] Y. Dodis, K. Pietrzak and D. Wichs, Key derivation without entropy waste, *Advances in CryptologyEUROCRYPT 2014, Lecture Notes in Comput. Sci., Springer, Heidelberg*, **8441** (2014), 93–110.

[10] Y. Dodis and Y. Yu, Overcoming weak expectations, *Theory of Cryptography, Springer*, (2013), 1–22.

[11] B. Gassend, D. Clarke, M. Van Dijk and S. Devadas, Delay-based circuit authentication and applications, *Proceedings of the 2003 ACM Symposium on Applied Computing*, (2003), 294–301.

[12] N. Gruzling, *Linear separability of the vertices of an n-dimensional hypercube*, Master's thesis, University of Northern British Columbia, 2008.

[13] J.-C. Hausmann, Counting polygon spaces, Boolean functions and majority games, Preprint, (2015), arXiv:1501.07553.

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

[15] M. Majzoobi, F. Koushanfar and M. Potkonjak, Lightweight secure PUFs, *Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design*, (2008), 670–673.

[16] S. Muroga, T. Tsuboi and C. R. Baugh, Enumeration of threshold functions of eight variables, *IEEE Transactions on Computers*, **100** (1970), 818–825.

[17] A. Rényi, On measures of entropy and information, *1961 Proc. 4th Berkeley Sympos. Math. Statist. and Prob., Univ. California Press, Berkeley, Calif*, **1** (1961), 547–561.

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

[19] A. Schaub, J.-L. Danger, S. Guilley and O. Rioul, An improved analysis of reliability and entropy for delay PUFs, *21st Euromicro Conference on Digital System Design, DSD 2018, Prague, Czech Republic*, (2018), 553–560.

[20] M. Skorski, Key derivation for squared-friendly applications: Lower bounds, *IACR Cryptology ePrint Archive*, **157** (2016).

[21] T. J. Stieltjes, Extrait d'une lettre adressée à M. Hermite, *Bulletin of Science and Mathematics, 2nd Series*, **13** (1889), 170–172.

[22] S. Tajik, E. Dietz, S. Frohmann, J.-P. Seifert, D. Nedospasov, C. Helfmeier, C. Boit and H. Dittrich, Physical characterization of arbiter PUFs, *International Workshop on Cryptographic Hardware and Embedded Systems*, (2014), 493–509.

[23] R. O. Winder, Single stage threshold logic, *Switching Circuit Theory and Logical Design, SWCT 1961. Proceedings of the Second Annual Symposium on*, (1961), 321–332.

[24] R. O. Winder, Enumeration of seven-argument threshold functions, *IEEE Transactions on Electronic Computers*, (1965), 315–325.

[25] M.-D. Mandel Yu and S. Devadas, Recombination of physical unclonable functions, *35th Annual GOMACTech Conference*, (2010).

[26] Y. A Zuev, Methods of geometry and probabilistic combinatorics in threshold logic, *Discrete Mathematics and Applications*, **2** (1992), 427–438.

Received November 2018; 1st revision December 2018; 2nd revision December 2018.

E-mail address: alexander.schaub@telecom-paris.fr
E-mail address: olivier.rioul@telecom-paris.fr
E-mail address: boutros@tamu.edu
E-mail address: jean-luc.danger@telecom-paris.fr
E-mail address: sylvain.guilley@telecom-paris.fr