On the Response Entropy of APUFs

Vincent Dumoulin, Wenjing Rao, and Natasha Devroye
Department of Electrical and Computer Engineering
University of Illinois Chicago
E-mail: vdumou2, wenjing, devroye @uic.edu

***Abstract*—A Physically Unclonable Function (PUF) is a hardware security primitive used for authentication and key generation. It takes an input bit-vector challenge and produces a single-bit response, resulting in a challenge-response pair (CRP). The truth table of all challenge-response pairs of each manufactured PUF should look different due to inherent manufacturing randomness, forming a digital fingerprint. A *PUF's entropy* (the entropy of all the responses, taken over the manufacturing randomness and uniformly selected challenges) has been studied before and is a challenging problem. Here we explore a related notion – the *response entropy*, which is the entropy of an arbitrary response given knowledge of one (and later two) other responses. This allows us to explore how knowledge of some CRP(s) impacts the ability to guess another response.**

The Arbiter PUF (APUF) is a well-known PUF architecture based on accumulated delay differences between two paths. In this paper, we obtain in closed form the probability mass function of any arbitrary response given knowledge of one or two other arbitrary CRPs for the APUF architecture. This allows us to obtain the conditional response entropy and then to define and obtain the size of the entropy bins (challenge sets with the same conditional response entropy) given knowledge of one or two CRPs. All of these results depend on the probability that two different challenge vectors yield the same response, termed the response similarity of those challenges. We obtain an explicit closed form expression for this. This probability depends on the statistical correlations induced by the PUF architecture together with the specific known and to-be-guessed challenges. As a by-product, we also obtain the optimal (minimizing probability of error) predictor of an unknown challenge given access to one (or two) challenges and the associated predictability.

***Index Terms*—Arbiter PUF, CRP correlation, entropy bins, expected entropy, response entropy.**

I. Introduction

Physically Unclonable Functions (PUFs) are circuits that can be integrated into chip designs to provide a low-cost digital “fingerprint”. They show promise as hardware security primitives for Internet of Things (IoT) devices that need low-power cryptographic frameworks for authenticated communication. PUF designs rely on randomness derived from many types of process variations, such as gate/wiring delays, designed to be the same across all chips, but inevitably differing due to random defects that occur during manufacturing [1], [2]. This results in an uncontrollable, unique function for each device, which is physically unclonable. With PUFs, the “fingerprint” of a device is not some stored bit-stream, but rather extracted from its unique input / output function, that may be “challenged” with an input vector $\mathbf{c} \in \{0,1\}^n$, which interacts with the random elements of a PUF instance to produce a “response”, $R_{\mathbf{c}} \in \{\pm 1\}$. The pair $(\mathbf{c}, R_{\mathbf{c}})$ is called a challenge-response pair (CRP). A subset of CRPs, usually randomly selected, can be used to uniquely identify or authenticate a device. There are two classes of PUFs. In weak PUFs the size of the truth table with respect to the number of random elements is small and hence must be kept secret as it is easily obtained via exhaustive evaluation; they are mostly used for key-generation in cryptography. Strong PUFs offer a huge number of CRPs, usually exponential in the number of physical random elements, and are primarily used in device authentication [1]. How good a PUF is is often measured by their response bias, the uniqueness, and the entropy of the PUFs, among other statistical metrics [3]–[5]. The arbiter PUF (APUF), a well known strong PUF architecture, is the focus of this paper, which is formally introduced in Section II.

A basic authentication framework for the APUF is as follows. During an enrollment phase, which takes place in a secure environment, a large set of random challenges are passed to the APUFs input and the outputs are recorded to create CRPs. This large set of CRPs (or a model of the PUF obtained via Machine Learning from this set) is then stored on the authentication server. During the authentication phase the APUF is no longer guaranteed to be in a secure environment. In order to authenticate a particular APUF according to various protocols [6]–[9], a small set of challenges are used to query the APUF by the server, so as to verify the APUF's identity.

CRP correlation: The PUF community has nearly always assumed the use of random CRP sets either in protocols or analysis of PUF metrics, or to build multi-bit responses. It has not deeply studied how challenges may be correlated, how exposing one challenge (e.g. to an attacker) may impact how predictable another challenge becomes.

We focus on presenting rigorous mathematical tools to understand how much knowledge of one or two challenges reveals about the remaining challenges in an arbiter PUF. We propose to measure this using the *response entropy*, and the *conditional response entropy*, or entropy of a response to a challenge given (in the conditional setting) knowledge of one or two other CRPs. This turns out to be a function of how statistically correlated the CRPs of an APUF are. We explicitly derive the response similarity $P[R_{\mathbf{c}'} = R_{\mathbf{c}}]$, which allows us to characterize the conditional response entropy and the associated "entropy bins" which contain all the challenges who have the same conditional entropy given knowledge of one (or two) CRPs.

Prior observations about the PUF CRP correlations: Our method is based on the analytical derivation of the probability that the responses to different challenges are the same. This probability has been observed experimentally and numerically in previous work. In particular, for APUFs, figures such as [10, Figure 6, simulated], [11, Figure 12], comments such as those in [12], and partial analytical derivations (non-closed form integrals) as in [13] focus on the how likely the responses to two challenges differing in one bit are to be the same, and plot the probability that the response changes as a function of the bit flip position (or two consecutive bit positions [13]). However, to the best of our knowledge, this bit flip probability as an explicit closed-form function of the challenges and responses has never been derived and is one of the contributions here. We furthermore provide closed-form expressions for *any* two arbitrarily correlated challenges $\mathbf{c}$ and $\mathbf{c}'$ and not only ones that change in one or two bit positions. We also propose explicit algorithms enumerating all challenges with a desired response similarity to a given one or two challenges.

Prior observations about the PUF entropy: In order for PUFs to be able to uniquely identify many devices, one hopes that the entropy of the PUFs manufactured in a particular architecture is large. This *PUF entropy*, or the entropy of all the responses (to all challenges) or a particular architecture has been studied before and is usually experimentally obtained, as it is difficult to analytically characterize.

In [14] they asked the question of how to select challenges to maximize the entropy (of the corresponding responses) of loop PUF outputs. They used an analytical model for the loop PUF that assumes Gaussian delay elements as justified in [15], and showed that n bits of entropy (equated with the randomness or hardness of predicting a response given no other responses) may be obtained from n challenges if and only if the challenges constitute a Hadamard code. They do not touch on the probability that two challenges result in the same response as a function of the challenges. Later work such as [16] focuses on estimating the probability distribution of certain kinds of PUFs composed of delay elements and finds the resulting Shannon entropy of the PUF is close to the max-entropy, which is asymptotically quadratic in the number of stages n. [17] presents a new approach for determining the min-entropy of a PUF based on convolving histograms. [18] analyzes the entropy of FPGA Lookup Table-based PUFs. None of these works focus on conditional *response* entropy estimation given knowledge of some challenges, but rather on estimating the overall PUF entropy. There is also work on the statistical analysis of PUFs [19], and characterizing the entropy of "strong" PUFs [19], [20] but these have been experimental and focus on the inherent qualities (bias, uniqueness and reliability) of PUFs, and do not focus on how challenges correlate the PUF responses.

Contributions: for APUF, we define and obtain:

• The *response similarity* between any pair of challenges $\mathbf{c}$ and $\mathbf{c}'$, $P[R_{\mathbf{c}'} = R_{\mathbf{c}}]$, often denoted by p. This response similarity is expressed as a function of the *similarity factor*, a function of the two challenges, and denoted as s. To the best of our knowledge, this is the first time that a closed form, analytical characterization of the probability that two or three challenges will produce the same response for an APUF. This underpins the Strict Avalanche Criterion (SAC) property for APUFs and yields an alternative to the Monte Carlo simulations often used in papers, e.g. [21] to simulate the probability that the response flips if for example one challenge input bit flips. In fact, our work provides a complete generalization of this single bit flip probability (where only one bit is flipped with respect to a base or anchor challenge) to the probability of flipping any number of bits with respect to an anchor challenge.

• The *similarity bins* of any anchor challenge $\mathbf{c}$, $B_s(p, \mathbf{c})$: for any given challenge (anchor) and a given response similarity p, we develop an efficient algorithm to find the set of all challenges that have the same given response similarity with respect to the anchor.

• The *response entropy* and the *conditional response entropy* of a challenge, conditioned on knowing one or two CRP(s). From this we are able to compute the conditional minimum response entropy and the conditional Shannon response entropy using the closed form expression for the probability that two or three challenges will produce the same response.

• The *entropy bins* of any anchor challenge or pair of challenges $\mathbf{c}$, $B_H(h, \mathbf{c})$: these are sets of challenges which all have the same response conditional entropy given $\mathbf{c}$. We are able to calculate the size of each entropy bin exactly. Such sets (entropy bins) can then be used to compute the expected conditional entropy of a response.

• The *expected conditional entropy*: knowing the size of each entropy bin allows us to calculate the expected conditional response entropy (not the conditional PUF entropy, which is more challenging [16]) when knowing one or two challenges.

• Our results yield an immediate **application:** finding the optimal predictor (minimizing the probability of error) of the response to one unknown challenge $\mathbf{c}'$ given the response to one (and later, extended to two) known challenge(s).

II. APUF architecture and notation

The APUF may be viewed as a Boolean function taking as input a challenge vector $\mathbf{c} := (c_1, c_2, \cdots c_n) \in \{0,1\}^n$ and outputting a response $R_{\mathbf{c}} \in \{\pm 1\}$. This response corresponds to the output of a race resolution element, latch, or arbiter, which detects which of two racing signals arrives first. The two signals traverse n multiplexers in series, and in the i-th stage, traverse the "parallel" paths (t_i, u_i) if the challenge bit $c_i = 0$, else traverse the "crossed" paths (r_i, s_i) if the challenge bit $c_i = 1$. The response then is -1 if the upper entrance to the arbiter arrives earlier than the lower, and is $+1$ otherwise - this is modeled by whether the final accumulated delay difference Δ_n is positive or negative at the entrances of the arbiter.

In the interest of modeling this behavior analytically, in stage $i \in [1, n]$ of the APUF the delays of the four possible paths taken (parallel or crossed) are modeled as four random delay elements, t_i, u_i, r_i and s_i, which are all i.i.d. normal random variables with mean μ and variance σ^2, denoted as $\sim \mathcal{N}(\mu, \sigma^2)$. they are always selected in fixed pairs, and since the response depends only on which signal arrives first, so on the relative delay difference between the two racing paths. The assumption that all have the same mean corresponds to the fact that the manufacturing process aims to produce identical MUXes, but due to process variation, they end up similar but not identical, with a small spread around the mean. Gaussians are good models for the inevitable manufacturing randomness, as justified in [15], [22]. Truncated Gaussians may be more suitable, but are less analytically tractable and not much better for the small variance of the manufacturing delay.

The challenge bit $c_i \in \{0, 1\}$ determines two factors: 1) which path pair (t_i, u_i in parallel, or r_i, s_i crossing) at stage i is chosen, and 2) the sign of the accumulated delay difference from all the previous stages before reaching i. The response $R_{\mathbf{c}}$ is expressed as the sign of the accumulated delay difference at the final stage Δ_n. In general, the accumulated delay Δ_i is recursively defined involving the challenge $\mathbf{c}$, the delay elements r_i, s_i, t_i, u_i of the current stage, and the accumulated delay difference at stage $i-1$, Δ_{i-1}:

$$R_{\mathbf{c}} = \text{sign}(\Delta_n) \in \{\pm 1\},$$

where Δ_n is computed recursively for $i \in [1, n]$, as

$$\Delta_i = \begin{cases} +\Delta_{i-1} + t_i - u_i, & \text{when } c_i = 0 \\ -\Delta_{i-1} + s_i - r_i, & \text{when } c_i = 1 \end{cases}, \quad \Delta_0 = 0.$$

The recursive formula is not easy for modeling and analysis, thus APUFs are frequently modeled by transforming the n-bit input vector $\mathbf{c}$ into another $n+1$ bit vector Φ, the $4n$ random variables into a vector $\mathbf{w}$ of size $n+1$. The response is then expressed as a linear threshold function of $\mathbf{\Phi}, \mathbf{w}$, in a non-recursive fashion as shown in [23]:

$$R_{\mathbf{\Phi}} = \text{sign}(\mathbf{\Phi} \cdot \mathbf{w}) = \text{sign}\left(\sum_{i=1}^{n+1} \phi_i w_i\right) \in \{\pm 1\}, \qquad (1)$$

where $\mathbf{\Phi} := (\phi_1, \phi_2, \cdots \phi_{n+1}) \in \{\pm 1\}^{n+1}$ is a vector depending solely on the challenge vector $\mathbf{c}$, and $\mathbf{w} := (w_1, w_2, \cdots w_{n+1}) \in \mathbb{R}^{n+1}$ is a vector depending only on the delay random variables, and $\cdot$ denotes the inner product:

$$\phi_i = \begin{cases} (-1)^{\sum_i^n c_k}, & 1 \leq i \leq n, \\ +1 & i = n+1 \end{cases} \qquad (2)$$

$$w_i = \begin{cases} (t_1 - u_1) - (s_1 - r_1), & i = 1 \\ \\ (t_{i-1} - u_{i-1}) + (s_{i-1} - r_{i-1}) \\ +(t_i - u_i) - (s_i - r_i), & 2 \leq i \leq n, \\ \\ (t_n - u_n) + (s_n - r_n), & i = n+1 \end{cases} \qquad (3)$$

This representation eases the analysis as it separates the $\mathbf{\Phi}$ (dependent only on the challenge) from $\mathbf{w}$ (dependent only on the random delay elements), and the output is now expressed as a linear threshold function. ML attacks against APUFs often exploit this representation.

This transformation renders the w_i's Gaussian i.i.d., all with the same variance except w_1 and w_{n+1} (which have half the variance as the others), as outlined in the following Lemma.

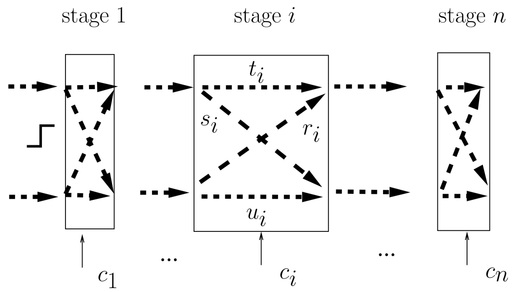


Fig. 1: Illustration of the delay elements

Lemma II.1. *For the arbiter PUF whose output is described by Eq.* (1)*, and using the transformation in Eq.* (3)*, then*

$$w_1 \sim \mathcal{N}(0, 4\sigma^2), \;\; w_i \sim \mathcal{N}(0, 8\sigma^2)$$
$$\textit{for } 2 \leq i \leq n, \;\; w_{n+1} \sim \mathcal{N}(0, 4\sigma^2),$$

and w_i, w_j are independent for $i \neq j$.

The transformation between the vectors $\mathbf{c}$ and $\mathbf{\Phi}$ is bijective. For the rest of the paper we transform challenges $\mathbf{c}, \mathbf{c}'$ into $\mathbf{\Phi}, \mathbf{\Phi}'$, which turns the APUF response into a linear threshold function, which is easier to deal with.

III. Response similarity and entropy

Our technical contribution comes from the succinct derivation of the following two notions, which are then (in the following subsection) used to calculate the conditional response entropy given knowledge of another CRP.

The **"response similarity"** between $\mathbf{\Phi}$ and $\mathbf{\Phi}'$, $P[R_{\mathbf{\Phi}'} = R_{\mathbf{\Phi}}]$, is the probability that they produce the same response. The probability is taken over the random generation of the delay elements, and NOT over the fixed challenges. The **"similarity bins"** of an "anchor" challenge $\mathbf{\Phi}$ are sets of challenges with equal response similarity to an anchor challenge.

The main theoretical results solve these two problems for an n-stage arbiter PUF (with minor modifications to any other PUF that may be modeled as an n-stage linear threshold function as long as the randomness in the stages are independent and Gaussian distributed):

- **Problem 1:** given challenge pair $(\mathbf{\Phi}, \mathbf{\Phi}')$, derive their response similarity $p = P[R_{\mathbf{\Phi}} = R_{\mathbf{\Phi}'}]$.
- **Problem 2:** given a CRP $(\mathbf{\Phi}, R_{\mathbf{\Phi}})$, optimally predict the target response $R_{\mathbf{\Phi}'}$ of another challenge $\mathbf{\Phi}'$, and derive the prediction accuracy (i.e. find $P[R_{\mathbf{\Phi}'} | R_{\mathbf{\Phi}}]$)
- **Problem 3:** given a challenge $\mathbf{\Phi}$ (an "anchor"), derive all the challenges with the same response similarity p to $\mathbf{\Phi}$, $B_s(p, \mathbf{\Phi})$, or the same accuracy $B_a(p, \mathbf{\Phi})$ (and later, the same entropy $B_H(p, \mathbf{\Phi})$).

Remark : The predictor works for a particular PUF instance that is not "abnormal" [24] i.e. there are no dominant or unusually large delay stages.

A. Response similarity and accuracy

1) Motivational example: Recall that $R_{\mathbf{\Phi}} = \text{sign}\left(\sum_{i=1}^{n+1} \phi_i w_i\right)$ and $R_{\mathbf{\Phi}'} = \text{sign}\left(\sum_{i=1}^{n+1} \phi_i' w_i\right)$, where $\phi_i = (-1)^{\sum_i^n (c_k)}$ and $\phi_i' = (-1)^{\sum_i^n (c_k')}$, with

$\phi_{n+1} = \phi'_{n+1} = 1$ for all $\mathbf{c}, \mathbf{c}'$ from equations (1) – (2). We now consider the following 3 challenges, among them the likelihood of some pair resulting in the same response (over an "average" PUF):

$$\begin{aligned} \mathbf{c} &= \texttt{(00000)} \quad \leftrightarrow \quad \Phi = \texttt{(+1,+1,+1,+1,+1,+1)} \\ \mathbf{c}' &= \texttt{(10000)} \quad \leftrightarrow \quad \Phi' = \texttt{(-1,+1,+1,+1,+1,+1)} \\ \mathbf{c}'' &= \texttt{(00001)} \quad \leftrightarrow \quad \Phi''' = \texttt{(-1,-1,-1,-1,-1,+1)} \end{aligned}$$

These challenges will lead to responses of the form (recalling that $\phi_{n+1} = 1$ always):

$$\begin{aligned} R_{\mathbf{\Phi}} &= \text{sign}(+w_1 + w_2 + w_3 + w_4 + w_5 + w_6) \\ R_{\mathbf{\Phi}'} &= \text{sign}(-w_1 + w_2 + w_3 + w_4 + w_5 + w_6) \\ R_{\mathbf{\Phi}''} &= \text{sign}(-w_1 - w_2 - w_3 - w_4 - w_5 + w_6) \end{aligned}$$

We can note the following: 1) $\mathbf{\Phi}$ and $\mathbf{\Phi}'$ are most likely to yield the same response because their corresponding $R_{\mathbf{\Phi}}$ and $R_{\mathbf{\Phi}'}$ differ only in w_1; 2) $\mathbf{\Phi}$ and $\mathbf{\Phi}''$ are very likely to yield the opposite responses, because they have all the w_i's in opposite signs, except for w_6. All of these observations can be confirmed by the following Theorem, where for the pair $(\mathbf{\Phi}, \mathbf{\Phi}')$ the similarity factor $s = S(\mathbf{\Phi}, \mathbf{\Phi}')$ is defined, and is a measure of how similar $\mathbf{\Phi}$ and $\mathbf{\Phi}'$ are.

2) Main Theorem: This section's main result is Theorem III.1, which shows the formal solution to Problem 1.

All proofs of Lemmas may be found in the Appendix; the main Theorem proof is shown in the text.

Theorem III.1 (Response similarity of APUFs.)**.** *For an APUF with a pair of challenges $\mathbf{\Phi}, \mathbf{\Phi}' \in \{\pm 1\}^{n+1}$ their response similarity (i.e., the probability of $R_{\mathbf{\Phi}} = R_{\mathbf{\Phi}'}$) is*

$$P[R_{\mathbf{\Phi}} = R_{\mathbf{\Phi}'}] = \frac{1}{2} + \frac{1}{\pi}\left[\arcsin\left(\frac{2S(\mathbf{\Phi}, \mathbf{\Phi}')}{n} - 1\right)\right]$$

where

$$S(\mathbf{\Phi}, \mathbf{\Phi}') := \frac{1}{2}\mathbf{1}_{\phi_1 = \phi'_1} + \sum_{i=2}^{n} \mathbf{1}_{\phi_i = \phi'_i} + \frac{1}{2}$$

is the "similarity factor" between $\mathbf{\Phi}$ and $\mathbf{\Phi}'$. This takes on values $\in [0, n]$ by steps of 0.5 depending on if $\phi_1 = \phi'_1$. Sometimes we drop the arguments and call it s (when evaluated, as a number) for brevity. It indicates how many bits are the "same" in $\mathbf{\Phi}$ and $\mathbf{\Phi}'$: with ϕ_1 and ϕ_{n+1} handled separately as they have half the weight as the other w_i's according to the APUF-specific transformation in (2).

Interpretation: you can see the response similarity (i.e., the probability of $R_{\mathbf{\Phi}} = R_{\mathbf{\Phi}'}$) as the expected number of times (in %) that the response to challenge $\mathbf{\Phi}'$ will be the same as the response to the anchor $\mathbf{\Phi}$ where the expectation is taken across multiple PUF instances. This is equivalent to the probability (over the PUF generation process) that for a given PUF, the two challenges will have the same response.

As an example of how to use this Theorem, consider the previous example challenges $\mathbf{\Phi}, \mathbf{\Phi}', \mathbf{\Phi}''$ from Section III-A1. For these, $S(\mathbf{\Phi}, \mathbf{\Phi}') = 4.5, P[R_{\mathbf{c}} = R_{\mathbf{c}'}] = \frac{1}{2} + \frac{1}{\pi}\left[\arcsin\left(\frac{9}{5} - 1\right)\right] \sim 0.8$, and $S(\mathbf{\Phi}, \mathbf{\Phi}'') = 1/2, P[R_{\mathbf{c}} = R_{\mathbf{c}'"}] = \frac{1}{2} + \frac{1}{\pi}\left[\arcsin\left(\frac{1}{5} - 1\right)\right] \sim 0.2$, aligning with the intuitive arguments before.

3) Proof of Theorem III.1: We can write $\mathbf{\Phi_1} := (\phi_{1,1}, \phi_{1,2}, \cdots, \phi_{1,n}, 1)$ and $\mathbf{\Phi_2} := (\phi_{2,1}, \phi_{2,2}, \cdots, \phi_{2,n}, 1)$. Recall that $R_{\mathbf{\Phi_1}} = \text{sign}\left(\sum_{i=1}^{n+1} \phi_{1,i} w_i\right)$ and $R_{\mathbf{\Phi_2}} = \text{sign}\left(\sum_{i=1}^{n+1} \phi_{2,i} w_i\right)$. Then, this proof follows using basics of probability and Lemma III.2 below. We see that

$$\begin{aligned} P[R_{\mathbf{\Phi_1}} = R_{\mathbf{\Phi_2}}] &= 2P[\Delta_n(\mathbf{\Phi_1}) > 0, \Delta_n(\mathbf{\Phi_2}) > 0] \\ &\overset{(a)}{=} \frac{1}{2} + \frac{1}{\pi}\left[\arcsin \rho_{\Delta_n(\mathbf{\Phi_1})\Delta_n(\mathbf{\Phi_2})}\right] \end{aligned}$$

where (a) follows by the multivariate Gaussian distribution orthant probabilities (Lemma III.2 below), and where ρ_{AB} is the correlation coefficient between random variables A and B:

$$\rho_{AB} := \frac{E[AB]}{\sqrt{\text{Var}(A)}\sqrt{\text{Var}(B)}}. \tag{4}$$

Let define the set S_{ij} indicates the set of indices for which $\mathbf{\Phi_i}$ and $\mathbf{\Phi_j}$ are equal (excluding index $n+1$):

$$S_{12} := \{i \in \{1, 2, \cdots n\} : \phi_{1,i} = \phi_{2,i}\}.$$

In the same way, it is possible to define the set D_{ij} of indices for which $\mathbf{\Phi_i}$ and $\mathbf{\Phi_j}$ are not equal/different:

$$D_{12} := \{i \in \{1, 2, \cdots n\} : \phi_{1,i} \neq \phi_{2,i}\}$$

So $\rho_{12} := \rho_{\Delta_n(\mathbf{\Phi_1})\Delta_n(\mathbf{\Phi_2})}$ may be calculated as

$$\begin{aligned} \rho_{12} &= \frac{E[\Delta_n(\mathbf{\Phi_1})\Delta_n(\mathbf{\Phi_2})]}{\sqrt{\text{Var}\Delta_n(\mathbf{\Phi_1})}\sqrt{\text{Var}\Delta_n(\mathbf{\Phi_2})}} \\ &= \frac{E[\Big(\sum_{i \in S_{12} \cup (n+1)} \phi_i w_i\Big)^2] - E[\big(\sum_{i \in D_{12}} \phi_i w_i\big)^2]}{\sqrt{4n\sigma^2}\sqrt{4n\sigma^2}} \\ &= \frac{\sum_{i \in S_{12} \cup (n+1)} E[|w_i|^2] - \sum_{i \in D_{12}} E[|w_i|^2]}{\sqrt{4n\sigma^2}\sqrt{4n\sigma^2}} \end{aligned}$$

as $\text{Var}(\Delta_n(\mathbf{\Phi_1})) = \text{Var}(\Delta_n(\mathbf{\Phi_2})) = 4n\sigma^2$. To further evaluate the expression in Eq. (5) due to the $(\Phi, \mathbf{w})$ transformation/notation, we note that $E[|w_i|^2] = 4\sigma^2$ for $i \in \{2, \cdots n\}$ but that $E[|w_i|^2] = 2\sigma^2$ for $i = 1, n{+}1$, by Lemma II.1. Since index $n+1$ is always in S_{12}, we need to determine whether index 1 should be in S_{12} or D_{12}.

If $\phi_{1,1} = \phi_{2,1}$ and hence index 1 also lies in S_{12}, we have

$$\rho_{12} = \frac{4\sigma^2 + 4\sigma^2(|(S_{12}| - 1) - |D_{12}|)}{4n\sigma^2} = \frac{|S_{12}| - |D_{12}|}{n}. \tag{5}$$

If $\phi_{1,1} = -\phi_{2,1}$ and hence index 1 is in D_{12}, we have

$$\rho_{12} = \frac{4\sigma^2 + 4\sigma^2(|S_{12}| - (|D_{12}| - 1))}{4n\sigma^2} = \frac{1 + |S_{12}| - |D_{12}|}{.} n$$

Combining this with the "similarity factor" notation, we obtain a succinct expression for ρ_{12}:

$$\rho_{12} = \frac{2S(\mathbf{\Phi_1}, \mathbf{\Phi_2})}{n} - 1 = \begin{cases} \frac{2|S_{12}|}{n} - 1 & \text{if } \phi_{1,1} = \phi_{2,1} \\ \frac{2|S_{12}|+1}{n} - 1 & \text{if } \phi_{1,1} \neq \phi_{2,1} \end{cases}.$$

Lemma III.2. *Let $X \sim \mathcal{N}(0,\sigma^2)$, $Y \sim \mathcal{N}(0,\sigma^2)$ with correlation coefficient $E[XY] = \rho_{xy}$. Then,*

$$P[X>0, Y>0] = \frac{1}{4} + \frac{\arcsin \rho_{xy}}{2\pi}.$$

4) Application of Theorem III.1: : **An “optimal predictor” $\widehat{R_{\mathbf{\Phi_2}}}(R_{\mathbf{\Phi_1}})$ to predict $R_{\mathbf{\Phi_2}}$ based on the known $(\mathbf{\Phi_1}, R_{\mathbf{\Phi_1}})$**

One immediate application of Theorem III.1 is to find the optimal predictor of the response $R_{\mathbf{\Phi_2}}$ to challenge $\mathbf{\Phi_2}$ once we know the response $R_{\mathbf{\Phi_1}}$ to challenge $\mathbf{\Phi_1}$. To find this, note that the conditional probability mass function

$$\begin{aligned} P[R_{\mathbf{\Phi_2}}|R_{\mathbf{\Phi_1}}] &= \frac{P[R_{\mathbf{\Phi_2}}, R_{\mathbf{\Phi_1}}]}{P[R_{\mathbf{\Phi_1}}]} = \frac{P[R_{\mathbf{\Phi_2}}, R_{\mathbf{\Phi_1}}]}{1/2} \\ &= \begin{cases} 2P[R_{\mathbf{\Phi_2}}=1, R_{\mathbf{\Phi_1}}=1] = P[R_{\mathbf{\Phi_1}} = R_{\mathbf{\Phi_2}}] & \text{if } R_{\mathbf{\Phi_2}} = R_{\mathbf{\Phi_1}} \\ 2P[R_{\mathbf{\Phi_2}}=1, R_{\mathbf{\Phi_1}}=-1] = 1 - P[R_{\mathbf{\Phi_1}} = R_{\mathbf{\Phi_2}}] & \text{O/W} \end{cases} \end{aligned}$$

may be derived immediately from Theorem III.1 and used to obtain the following *optimal predictor* for $R_{\mathbf{\Phi_2}}$ based on $R_{\mathbf{\Phi_1}}$, written as $\widehat{R_{\mathbf{\Phi_2}}}(R_{\mathbf{\Phi_1}})$:

Corollary III.2.1. *The optimal predictor for the response $R_{\mathbf{\Phi_2}}$ to an arbiter PUF challenged with $\mathbf{\Phi_2}$ given knowledge of the CRP $(\mathbf{\Phi_1}, R_{\mathbf{\Phi_1}})$ is,*

$$\begin{aligned} \widehat{R_{\mathbf{\Phi_2}}}(R_{\mathbf{\Phi_1}}) &= \arg \max_{R_{\mathbf{\Phi_2}} \in \{\pm 1\}} P[R_{\mathbf{\Phi_2}}|R_{\mathbf{\Phi_1}}] \\ &= \begin{cases} R_{\mathbf{\Phi_1}} & \textit{if } P[R_{\mathbf{\Phi_1}} = R_{\mathbf{\Phi_2}}] > 1 - P[R_{\mathbf{\Phi_1}} = R_{\mathbf{\Phi_2}}] \\ -R_{\mathbf{\Phi_1}} & \textit{if } P[R_{\mathbf{\Phi_1}} = R_{\mathbf{\Phi_2}}] < 1 - P[R_{\mathbf{\Phi_1}} = R_{\mathbf{\Phi_2}}] \end{cases} \end{aligned} \tag{6}$$

where ties (when the two are exactly equal) may be broken arbitrarily. The prediction accuracy given by

$$\max\{P[R_{\mathbf{\Phi_1}} = R_{\mathbf{\Phi_2}}], 1 - P[R_{\mathbf{\Phi_1}} = R_{\mathbf{\Phi_2}}]\}$$

with

$$\begin{aligned} P[R_{\mathbf{\Phi_1}} = R_{\mathbf{\Phi_2}}] &= \frac{1}{2} + \frac{1}{\pi}\left[\arcsin\left(\frac{2s}{n} - 1\right)\right] \\ 1 - P[R_{\mathbf{\Phi_1}} = R_{\mathbf{\Phi_2}}] &= \frac{1}{2} - \frac{1}{\pi}\left[\arcsin\left(\frac{2s}{n} - 1\right)\right] \end{aligned}$$

where $s = S(\mathbf{\Phi_1}, \mathbf{\Phi_2})$.

As an example application of Corollary III.2.1, consider $\mathbf{c} = 00000$ (as this is what is physically input to the PUF we present it in this notation, but all calculations are done by transforming $\mathbf{c}$ to $\mathbf{\Phi}$ first), with $R_{\mathbf{c}} = +1$ and say we wish to predict $R_{\mathbf{c}'}$, the response to $\mathbf{c}' = 00110$. Since $P[R_{\mathbf{c}} = R_{\mathbf{c}'}] \sim 0.7 > (1-0.7)$, we should guess that $R_{\mathbf{c}'}$ is also equal to $+1$ and this has a probability of 0.7 of being correct.

This predictor maximizes the probability of correctly guessing (in one guess) the value of $R_{\mathbf{\Phi}'}$ based on knowledge of $R_{\mathbf{\Phi}}$, hence we term the predictor in (6) the *optimal predictor* in this sense. Here, if we have one guess for $R_{\mathbf{\Phi}'}$ based on knowledge of $R_{\mathbf{\Phi}}$ this corresponds to guessing the $\widehat{R_{\mathbf{\Phi}'}}(R_{\mathbf{\Phi}})$ that maximizes $P[R_{\mathbf{\Phi}'}|R_{\mathbf{\Phi}}]$ as in (6).

B. Entropy

Recall the definitions of entropy of random variable (or vector) X with probability mass function $P_X(x)$ taking on values $x \in \mathcal{X}$ and conditional entropy of random variable X given random variable Y (with joint distribution $P_{X,Y}(x,y)$ taking on values $x, y \in \mathcal{X} \times \mathcal{Y}$ [25]:

$$\begin{aligned} H(X) &:= -\sum_{x\in\mathcal{X}} p_X(x)\log(p_X(x)) \\ H(X|Y=y) &:= -\sum_{x\in\mathcal{X}} p_{X|Y}(x|y)\log(p_{X|Y}(x|y)) \\ H(X|Y) &:= -\sum_{y\in\mathcal{Y}} p_Y(y) H(X|Y=y) \\ &= -\sum_{x\in\mathcal{X}, y\in\mathcal{Y}} p_{X,Y}(x,y)\log\left(\frac{p_{X,Y}(x,y)}{p_Y(y)}\right) \end{aligned}$$

We define the **PUF entropy** $H(\bigcup_{\mathbf{\Phi}} R_{\mathbf{\Phi}})$ as the entropy of the vector of responses $\bigcup_{\mathbf{\Phi}} R_{\mathbf{\Phi}}$, recalling that each response is a binary random value. Note that the joint distribution $P(\bigcup_{\mathbf{\Phi}} R_{\mathbf{\Phi}})$ is hard to capture analytically as orthant probabilities of Gaussian random vectors are generally unsolved for vectors of dimensions greater than 3, necessitating estimates or bounds on this. The **conditional PUF entropy** is the PUF entropy knowing one CRP: $H(\bigcup_{\mathbf{\Phi}} R_{\mathbf{\Phi}}|R_{\mathbf{\Phi_1}})$, which is equally hard to obtain given our inability to characterize the Gaussian orthant probabilities beyond dimension 3.

We thus study what we call **response entropy**, i.e. the entropy of one response $H(R_{\mathbf{\Phi}})$ and the **conditional response entropy** $H(R_{\Phi_2}|R_{\Phi_1})$ given knowledge of one CRP $(\mathbf{\Phi}, R_{\mathbf{\Phi}})$. Both the response entropy and the conditional response entropy pertain to the entropy of one binary response and hence take on values between 0 and 1. For the conditional response entropy $H(R_{\Phi_2}|R_{\Phi_1})$, this value will depend on how correlated Φ_1 and Φ_2 are, something we characterized precisely before using the response similarity. We can use the chain rule to link all those different entropies:

$$\begin{aligned} H\left(\bigcup_{\mathbf{\Phi}} R_{\mathbf{\Phi}}\right) &= \sum_{i=1}^{n} H(R_{\mathbf{\Phi}_i}|R_{\mathbf{\Phi}_1}, \cdots, R_{\mathbf{\Phi}_{i-1}}) \\ &= H(R_{\mathbf{\Phi}_1}) + H(R_{\mathbf{\Phi}_2}|R_{\mathbf{\Phi}_1}) + H(R_{\mathbf{\Phi}_3}|R_{\mathbf{\Phi}_2}, R_{\mathbf{\Phi}_1}) + \cdots \end{aligned}$$

This work calculates the first three terms (response entropies) exactly for any choice of $\mathbf{\Phi_1}, \mathbf{\Phi_2}, \mathbf{\Phi_3}$. The left term is the overall PUF entropy which is challenging to calculate.

All of the above definitions we presented the Shannon-entropy definition, but these definitions can be equivalently modified for the min-entropy. The Min-entropy of a probability mass function (or conditional probability mass function) is defined as the log of the most likely outcome. When the response $R_{\mathbf{\Phi_1}} = r_{\mathbf{\Phi_1}}$ is known, the most likely conditional entropy will be given by the probability that you guess $R_{\mathbf{\Phi_2}}$ (for a new, never before seen challenge $\mathbf{\Phi_2}$) correctly in one go. The min entropy is simply defined as

$$H_{\min}(R_{\mathbf{\Phi_2}}|R_{\mathbf{\Phi_1}} = r_{\mathbf{\Phi_1}}) = -\log\max\{p_s, 1-p_s\}.$$

Min-entropy is often used when considering worst-case scenarios, ensuring that even if the distribution is not uniform, the system remains secure. Shannon entropy is more commonly used when analyzing average-case scenarios. In the context of hardware security primitives, min-entropy may be preferred because it helps assess the worst-case security of the PUF when one or two CRPs have been revealed.

Without any CRP exposure, the response entropy $P[R_\Phi = 1] = P[R_\Phi = -1] = \frac{1}{2}$ so Min-response-entropy and Shannon response entropy both equal 1 bit. For arbitrary challenge bit vectors $\mathbf{\Phi_1}, \mathbf{\Phi_2}$ the a conditional response entropy becomes:

$$
\begin{aligned}
H(R_{\mathbf{\Phi_2}} | R_{\mathbf{\Phi_1}} = r_{\mathbf{\Phi_1}}) = &- \sum_{r_{\mathbf{\Phi_2}} \in \{\pm 1\}} P[R_{\mathbf{\Phi_2}} = r_{\mathbf{\Phi_2}} | R_{\mathbf{\Phi_1}} = r_{\mathbf{\Phi_1}}] \\
&\cdot \log_2(P[R_{\mathbf{\Phi_2}} = r_{\mathbf{\Phi_2}} | R_{\mathbf{\Phi_1}} = r_{\mathbf{\Phi_1}}]) \\
\overset{(a)}{=} &-p_s \log p_s - (1-p_s)\log(1-p_s),
\end{aligned}
$$

where (a) follows when we define $p_s = P[R_{\mathbf{\Phi_2}} = 1 | R_{\mathbf{\Phi_1}} = 1]$ or $p_s = P[R_{\mathbf{\Phi_2}} = -1 | R_{\mathbf{\Phi_1}} = -1]$ (depending on what value $r_{\mathbf{\Phi_1}} \in \{\pm 1\}$ takes on) and hence $1 - p_s = P[R_{\mathbf{\Phi_2}} = -1 | R_{\mathbf{\Phi_1}} = 1]$ or $1 - p_s = P[R_{\mathbf{\Phi_2}} = 1 | R_{\mathbf{\Phi_1}} = -1]$.

Figure 2 illustrates the similarity factors and their corresponding response similarity $p_s = \frac{1}{2} + \frac{1}{\pi}\arcsin\left(\frac{2s}{n} - 1\right)$ as a function of s for a 32-bit APUF. The x-axis shows the similarity factor $s = S(\mathbf{\Phi_1}, \mathbf{\Phi_2})$ between the known (say $\mathbf{\Phi_1}$) and the unknown (say $\mathbf{\Phi_2}$) challenges. The green points p_s shows the response similarity ranging from 0 to 1, corresponding to the y-axis on the left. We plot also both the Shannon and min entropies of a challenge's response when we know 1 challenge as a function of s. As we can see, this will depend on the correlation ρ_{12} between the challenge whose response we know, $\mathbf{\Phi_1}$, and the one whose response we wish to guess, $\mathbf{\Phi_2}$. If they are uncorrelated, i.e. $\rho_{12} = 0$ (or s roughly equal to $n/2$), then the response entropy remains optimal at 1 bit; the more correlated or anti-correlated they become, the more this drops, and more dramatically so for the min entropy.

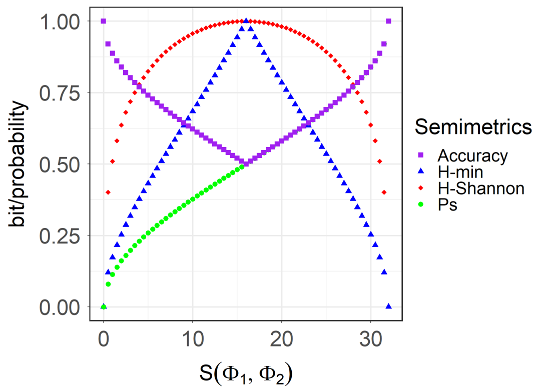


Fig. 2: Accuracy, probability of same, Min-entropy and Shannon entropy of a second challenge response $\mathbf{\Phi_2}$ when we know 1 challenge $\mathbf{\Phi_1}$'s response, as a function of the $S(\mathbf{\Phi_1}, \mathbf{\Phi_2})$. Non-integer values of $S(\mathbf{\Phi_1}, \mathbf{\Phi_2})$ occur when $\phi_{1,1} \neq \phi_{2,1}$.

IV. Similarity, accuracy and entropy Bins

Construction of Bins: Given an n-bit challenge $\mathbf{\Phi_1}$ (the *"anchor"*), we now show how to construct various sets containing all the challenges respecting a "semimetric" to the anchor. We can intuitively view a bin B of challenges as a sphere of challenges of radius r that all have the same probability, a prediction accuracy or conditional entropy given the anchor challenge. This radius can be

- the response similarity: *"similarity bins"* written $B(s, \mathbf{\Phi_1}) := \{\mathbf{\Phi_2} | S(\mathbf{\Phi_1}, \mathbf{\Phi_2}) = s\}$ or equivalently $B(p, \mathbf{\Phi_1}) := \{\mathbf{\Phi_2} | P[R_{\mathbf{\Phi_1}} = R_{\mathbf{\Phi_2}}] = p\}$ are the bins containing all the challenges that have the same response similarity to the anchor. For sake of the convenience, we will use the first definition $B(s, \mathbf{\Phi_1})$.
- the prediction accuracy: *"accuracy bins"* written $B_a(a, \mathbf{\Phi_1}) := \{\mathbf{\Phi_2} | \max\{P[R_{\mathbf{\Phi_1}} = R_{\mathbf{\Phi_2}}], 1 - P[R_{\mathbf{\Phi_1}} = R_{\mathbf{\Phi_2}}]\} = a\}$ have all the challenges which can be predicted with the same accuracy given the anchor challenge.
- the entropy (Shannon or min): *"entropy bins"* written $B_H(h, \mathbf{\Phi_1}) := \{\mathbf{\Phi_2} | H(R_{\mathbf{\Phi_2}} | R_{\mathbf{\Phi_1}} = r_{\mathbf{\Phi_1}}) = h\}$ all have the same conditional response entropy given the anchor challenge.

These bins B can be derived using the probability $P[R_{\mathbf{\Phi_1}} = R_{\mathbf{\Phi_2}}]$ which relies on the similarity factor. From a specific value of your "semimetric" you can derive the similarity factor(s) and then use algorithm 1 (Appendix A).

According to Theorem III.1, there are only $2n$ distinct response similarities thus for a given anchor challenge, all the challenges (including the anchor itself) will be partitioned into a total of $2n$ disjoint similarity bins. Intuitively, most of the challenges are in the "uncorrelated" bin of an anchor, i.e., $B(p \approx 0.5, \mathbf{\Phi})$.

First, consider the trivial case $B(p = 1, \mathbf{\Phi}) = B(s = n, \mathbf{\Phi})$: among all the 2^n challenges, the one with highest response similarity (100%) is the anchor itself and itself only, thus $B(n, \mathbf{\Phi}) = \{\mathbf{\Phi}\}$, and its size is 1.

Next, consider the similarity bin with the highest $p < 1$: this would be $B(p, \mathbf{\Phi}) = B(s = n - 1/2, \mathbf{\Phi})$, where $p = \frac{1}{2} + \frac{1}{\pi}\arcsin\left(\frac{2s}{n} - 1\right)$. For a challenge $\mathbf{\Phi'}$ to be in this set, it needs to satisfy $S(\mathbf{\Phi'}, \mathbf{\Phi}) = n - 1/2$ according to Theorem III.1. This can only be achieved by making $\phi'_1 = -\phi_1$, while keeping all other $\phi'_i = \phi_i, i \in [2, n+1]$ – as w_1 contributes half the weight as the other $w_i, i \in [2, n]$, while w_{n+1} always is positive (as $\phi_{n+1} = +1$). Thus, this will constitute $B(s = n - 1/2, \mathbf{\Phi})$, a set with only one element.

Following the same argument, a challenge $\mathbf{\Phi'} \in B(s = n - 1, \mathbf{\Phi})$ must satisfy $\phi'_1 = \phi_1$, with a single $\phi'_i = -\phi_i$ among $i \in [2, n]$. Thus the size of this bin $|B(n - 1, \mathbf{\Phi})| = n - 1$. Similarly, a challenge $\mathbf{\Phi'} \in B(s = n - 3/2, \mathbf{\Phi})$ can be derived by making $\phi'_1 = -\phi_1$, and making sure only a single $\phi'_i = -\phi_i$ among $i \in [2, n]$. The size of the bin is again $n - 1$.

Essentially, to obtain a challenge $\mathbf{\Phi'} \in B(s, \mathbf{\Phi})$, one needs to select $\lfloor s \rfloor$ among the anchor's $\phi_i, i \in [2, n]$ to flip their signs to form the ϕ'_i's. Whether $\phi'_1 = \phi_1$ or $-\phi_1$ depends on whether s is an integer or not.

Size of Similarity Bins: Algorithm 1 summarizes the general method for deriving the similarity bin $B(s, \mathbf{\Phi})$. The complexity is linear in the size of the similarity bin to be derived. From Algorithm 1, we can derive the size of a similarity bin as follows:

Corollary IV.0.1. *The size of the similarity bin $B(s, \mathbf{\Phi})$, for*

$s \in [1:n]$, *for an APUF of length* n *is*

$$\begin{aligned} |B(s,\mathbf{\Phi})| &= \binom{n-1}{s} && \textit{if } \phi_1 = \phi_1' \\ &= \binom{n-1}{\lfloor s-1 \rfloor} && \textit{if } \phi_1 \neq \phi_1' \end{aligned}$$

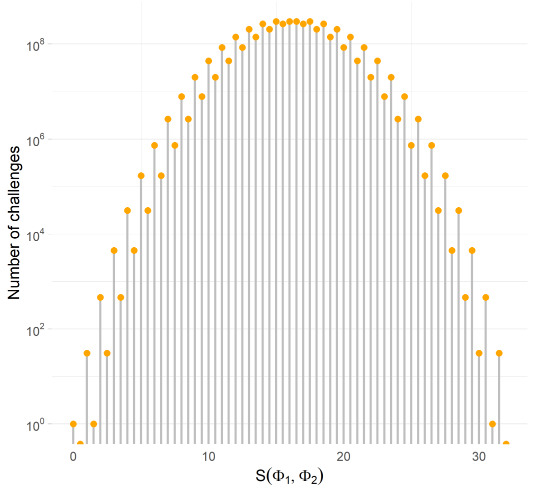


Fig. 3: Similarity bin sizes (number of challenges) in function of $S(\mathbf{\Phi_1}, \mathbf{\Phi_2})$. Not integer values of S means that $\phi_{1,1} \neq \phi_{2,1}$.

Figure 3 shows the size of each similarity bin, $|B(s,\mathbf{\Phi})|$, obtained from Corollary IV.0.1 on the right y-axis.

Expected conditional entropy: From Figures 2 and 3 we have all the information needed to compute the expected conditional response entropy or accuracy of a response given one CRP. We know the specific value of the response entropy for each possible known challenge as well as the exact number of challenges in an entropy bin, i.e the exact number of challenges such that $H(R_{\mathbf{\Phi'}}|R_{\mathbf{\Phi}} = r_{\mathbf{\Phi}}) = h$ since $B(h,\mathbf{\Phi})$ is the union of two disjoints $B(s,\mathbf{\Phi})$. Defining $h_{s,\mathbf{\Phi}}$ as $h_{s,\mathbf{\Phi}} = H(R_{\mathbf{\Phi'}}|R_{\mathbf{\Phi}} = r_{\mathbf{\Phi}})$ with $\mathbf{\Phi'} \in B(s,\mathbf{\Phi})$, the expected response entropy becomes the weighted average of the response entropies, weighted by the number of challenges in each bin as below:

$$\bar{H}(R_{\mathbf{\Phi'}}|R_{\mathbf{\Phi}} = r_{\mathbf{\Phi}}) = \frac{1}{2^n - 1} \sum_{s \in \{0, \frac{1}{2}, 1, \cdots, n - \frac{1}{2}\}} h_{s,\mathbf{\Phi}} \cdot |B(s,\mathbf{\Phi})|$$

The expected accuracy (of estimating a next challenge given knowledge of one CRP) can be similarly calculated as a weighted sum. The expected conditional min and Shannon entropies and the expected conditional accuracy are calculated in Table I for different PUF lengths. Note that this does not depend on the variance σ^2 of the delay element generation process.

V. Scalability to more known CRPs

All conditional response entropy and optimal predictor results so far have assumed a single anchor: i.e., *one* CRP $(\mathbf{\Phi}, R_{\mathbf{\Phi}})$ to be known. The natural next question is how to obtain optimal predictors and conditional response entropies with *multiple* anchors. This is based on finding the conditional probability mass function, which, at its core, depends on the availability of closed form expressions for the orthant probabilities of jointly Gaussian random variables. While this is known for Gaussian vectors of dimensions 2 (useful for one known, one target challenge) and 3 (useful for two known, and one target challenge), it remains unknown for larger dimensions [26], and hence there is little hope for *closed form* solutions for higher dimensions, i.e. optimally predicting the response to a target challenge given knowledge of 3 or more known CRPs. Numerical approximations are left for future work.

	n=32	n=64	n=128
H-min	0.8747	0.9112	0.9369
H-Shannon	0.9900	0.9952	0.9977
Accuracy	0.5465	0.5323	0.5226

TABLE I: Expected conditional entropy and expected accuracy of a challenge's response knowing one CRP for APUFs of different length (number of stages n).

A. Optimal predictor

When $(\mathbf{\Phi_1}, R_{\mathbf{\Phi_1}})$ and $(\mathbf{\Phi_2}, R_{\mathbf{\Phi_2}})$ are known and we wish to predict the response $R_{\mathbf{\Phi_3}}$ to a third challenge $\mathbf{\Phi_3}$, one naive approach may be to use our previous know-one-predict-one predictor to predict $R_{\mathbf{\Phi_3}}$ based on the response of the "most correlated" challenge to $\mathbf{\Phi_3}$, i.e. based on the response predicted by either $\mathbf{\Phi_1}$ or $\mathbf{\Phi_2}$ We will show that this is in fact provably optimal. This naive strategy can be extended to knowing more than 2 CRPs to predict another, but we are unable to prove optimality, which essentially stems from Gaussian orthant probabilities being unknown for more than 3 dimensions.

For knowing two CRPs and predicting a third, the optimal predictor takes on the following form [27]:

$$\widehat{R_{\mathbf{\Phi_3}}} = \arg\max P(R_{\mathbf{\Phi_3}}|R_{\mathbf{\Phi_1}}, R_{\mathbf{\Phi_2}}).$$

Hence, to obtain this optimal predictor, we need to obtain the conditional probability mass functions $P(R_{\mathbf{\Phi_3}}|R_{\mathbf{\Phi_1}}, R_{\mathbf{\Phi_2}})$. This involves finding the eight values

$$P(R_{\mathbf{\Phi_3}} = r_3|R_{\mathbf{\Phi_1}} = r_1, R_{\mathbf{\Phi_2}} = r_2), \quad r_i \in \{\pm 1\}.$$

We only need the four values $P(R_{\mathbf{\Phi_3}} = -1|R_{\mathbf{\Phi_1}} = r_1, R_{\mathbf{\Phi_2}} = r_2)$ from which we can find the others as $P(R_{\mathbf{\Phi_3}} = 1|R_{\mathbf{\Phi_1}} = r_1, R_{\mathbf{\Phi_2}} = r_2) = 1 - P(R_{\mathbf{\Phi_3}} = -1|R_{\mathbf{\Phi_1}} = r_1, R_{\mathbf{\Phi_2}} = r_2)$. By definition, we have that

$$\begin{aligned} &P(R_{\mathbf{\Phi_3}} = r_3|R_{\mathbf{\Phi_1}} = r_1, R_{\mathbf{\Phi_2}} = r_2) \\ &= \frac{P(R_{\mathbf{\Phi_3}} = r_3, R_{\mathbf{\Phi_1}} = r_1, R_{\mathbf{\Phi_2}} = r_2)}{P(R_{\mathbf{\Phi_1}} = r_1, R_{\mathbf{\Phi_2}} = r_2)}. \end{aligned} \tag{7}$$

Recall that $R_{\mathbf{\Phi}} = \text{sign}(\Delta_n(\mathbf{\Phi}))$, and that Δ_n is a Gaussian random variable, as it is the sum of Gaussian random variables. To obtain (7) we thus need only calculate $P(R_{\mathbf{\Phi_3}}, R_{\mathbf{\Phi_1}}, R_{\mathbf{\Phi_2}})$ and $P(R_{\mathbf{\Phi_1}}, R_{\mathbf{\Phi_2}})$. These may both be obtained by noting that since $R_{\mathbf{\Phi}} = \text{sign}(\Delta_n(\mathbf{\Phi}))$ are signs of zero mean, equal variance Gaussian random variables, these probabilities amount to the orthant probabilities of jointly Gaussian random variables, i.e.

$$\begin{aligned}
&P(R_{\boldsymbol{\Phi_3}} = 1 | R_{\boldsymbol{\Phi_1}} = 1, R_{\boldsymbol{\Phi_2}} = 1) \\
&= \frac{P(R_{\boldsymbol{\Phi_3}} = 1, R_{\boldsymbol{\Phi_2}} = 1, R_{\boldsymbol{\Phi_1}} = 1)}{P(R_{\boldsymbol{\Phi_1}} = 1, R_{\boldsymbol{\Phi_2}} = 1)} \\
&= \frac{P(\Delta_n(\boldsymbol{\Phi_3}) > 0, \Delta_n(\boldsymbol{\Phi_2}) > 0, \Delta_n(\boldsymbol{\Phi_1}) > 0)}{P(\Delta_n(\boldsymbol{\Phi_1}) > 0, \Delta_n(\boldsymbol{\Phi_2}) > 0))}.
\end{aligned}$$

If X, Y, Z are zero mean Gaussian random variables, then

$$\begin{aligned}
&P(X > 0, Y > 0, Z > 0) \\
&= \frac{1}{8} + \frac{1}{4\pi}\left[\arcsin \rho_{XY} + \arcsin \rho_{XZ} + \arcsin \rho_{XZ}\right] \\
&P(X > 0, Y > 0) = \frac{1}{4} + \frac{1}{2\pi}\left[\arcsin \rho_{XY}\right] \\
&P(X > 0 | Y > 0, Z > 0) \\
&= \frac{1}{2}\left[1 + \frac{\arcsin \rho_{XY} + \arcsin \rho_{XZ}}{\frac{\pi}{2} + \arcsin \rho_{YZ}}\right],
\end{aligned}$$

Letting ρ_{ij} be the correlation coefficient between $\Delta_n(\boldsymbol{\Phi_i})$ and $\Delta_n(\boldsymbol{\Phi_j})$ (which we recall are Gaussian random variables) we obtain the following:

Theorem V.1. *For* $r_1, r_2 \in \{\pm 1\}$,

$$\begin{aligned}
&P(R_{\boldsymbol{\Phi_3}} = 1 | R_{\boldsymbol{\Phi_1}} = r_1, R_{\boldsymbol{\Phi_2}} = r_2) \\
&= \frac{1}{2}\left[1 + \frac{r_1 \arcsin \rho_{13} + r_2 \arcsin \rho_{23}}{\frac{\pi}{2} + r_1 r_2 \arcsin \rho_{12}}\right]
\end{aligned} \tag{8}$$

and we can obtain $P(R_{\boldsymbol{\Phi_3}} = -1 | R_{\boldsymbol{\Phi_1}} = r_1, R_{\boldsymbol{\Phi_2}} = r_2) = 1 - P(R_{\boldsymbol{\Phi_3}} = 1 | R_{\boldsymbol{\Phi_1}} = r_1, R_{\boldsymbol{\Phi_2}} = r_2)$.

For clarity, we detail the quantities needed:

$$\begin{aligned}
&|S_{ij}| := \#\text{ indices } l \in \{1, 2, \cdots n\} \text{ for which } \phi_{i,l} = \phi_{j,l} \\
&\rho_{ij} = \begin{cases} \frac{2|S_{ij}|}{n} - 1 & \text{if } \phi_{i,1} = \phi_{j,1} \\ \frac{2|S_{ij}| + 1}{n} - 1 & \text{if } \phi_{i,1} \neq \phi_{j,1} \end{cases}.
\end{aligned}$$

From the above equation (8) we notice a few things: 1) first, the prediction accuracy depends on the actual response values r_1, r_2 and how they interact with the correlation coefficients. For example, for good prediction accuracy, you want the second term inside the brackets to be close to 1 or -1 (and hence the overall prediction being close to either 1 or 0). Poor accuracy corresponds to the second term being close to 0 (and hence the overall prediction being around 0.5). For good accuracy you want $r_1 \arcsin \rho_{13}$ and $r_2 \arcsin \rho_{23}$ to align or point in the same direction, both adding to a positive $+1$ or a negative -1, then it will be easy to predict the third r_3 as it will likely be in the same direction as the others. If the two challenge responses and correlation coefficients contradict each other, i.e. if $r_1 \arcsin \rho_{13} = -r_2 \arcsin \rho_{23}$ then the accuracy will be poor. The denominator also matters: if it becomes close to 0, or if $r_1 r_2 \arcsin \rho_{12}$ is close to $-\pi/2$ (could happen if $r_1 = r_2 = +1$ but $\arcsin \rho_{12} = -\pi/2$ which means that $\rho_{12} = -1$ and the challenges are statistically anti-correlated yet produced the same sign – virtually impossible), then the prediction accuracy is close to $\frac{1}{2}$.

The optimal predictor of $R_{\boldsymbol{\Phi_3}}$ given knowledge of $R_{\boldsymbol{\Phi_2}}$ and $R_{\boldsymbol{\Phi_1}}$ first calculates ρ_{12}, ρ_{13} and ρ_{23}. From this, and $R_{\boldsymbol{\Phi_2}}$ and $R_{\boldsymbol{\Phi_1}}$ we select $R_{\boldsymbol{\Phi_3}}$ as:

$$\widehat{R_{\boldsymbol{\Phi_3}}} = \arg \max_{R_{\boldsymbol{\Phi_3}} \in \{\pm 1\}} P(R_{\boldsymbol{\Phi_3}} | R_{\boldsymbol{\Phi_1}}, R_{\boldsymbol{\Phi_2}}).$$

The optimal prediction accuracy is then given by

$$\begin{aligned}
\text{Prediction accuracy} = \max\big\{&P[R_{\boldsymbol{\Phi_3}} = 1 | R_{\boldsymbol{\Phi_1}} = r_1, R_{\boldsymbol{\Phi_2}} = r_2], \\
&P[R_{\boldsymbol{\Phi_3}} = -1 | R_{\boldsymbol{\Phi_1}} = r_1, R_{\boldsymbol{\Phi_2}} = r_2]\big\}
\end{aligned}$$

which may be re-written as a value A as:

$$\begin{aligned}
A = \max\Big\{&\frac{1}{2}\left[1 + \frac{r_1 \arcsin \rho_{13} + r_2 \arcsin \rho_{23}}{\pi/2 + r_1 r_2 \arcsin \rho_{12}}\right], \\
&1 - \frac{1}{2}\left[1 + \frac{r_1 \arcsin \rho_{13} + r_2 \arcsin \rho_{23}}{\pi/2 + r_1 r_2 \arcsin \rho_{12}}\right]\Big\}.
\end{aligned}$$

This shows that the optimal predictor may be obtained in closed form for the arbiter PUF when two CRPs are known. In fact the optimal predictor has a simple form: if you know 2 challenges and want to predict a third, simply select the closest challenge to the anchor (the most correlated) and use the know-one-predict-one predictor. In other words, the response of the third challenge is the response of closest of the two challenges times the sign of its correlation:

$$\begin{aligned}
&P(R_{\boldsymbol{\Phi_3}} = 1 | R_{\boldsymbol{\Phi_1}} = r_1, R_{\boldsymbol{\Phi_2}} = r_2) \gtrless 0.5 \\
&\Leftrightarrow \frac{1}{2}\left[1 + \frac{r_1 \arcsin \rho_{13} + r_2 \arcsin \rho_{23}}{\frac{\pi}{2} + r_1 r_2 \arcsin \rho_{12}}\right] \gtrless 0.5 \\
&\Leftrightarrow \frac{r_1 \arcsin \rho_{13} + r_2 \arcsin \rho_{23}}{c} \gtrless 0 \\
&\Leftrightarrow r_1 \arcsin \rho_{13} + r_2 \arcsin \rho_{23} \gtrless 0
\end{aligned}$$

$$\begin{aligned}
&\Leftrightarrow r_1 \rho_{13} + r_2 \rho_{23} \gtrless 0 \\
&\Leftrightarrow R_{\boldsymbol{\Phi_3}} = r_i \cdot \text{sign}(\rho_{i3}) \text{ with } i = \arg\max\{|\rho_{13}|, |\rho_{23}|\}
\end{aligned}$$

B. Entropy

The conditional response entropy given two CRPs also depends only on the conditional probability mass functions, this is easily obtained once we have the conditional distributions, as:

$$\begin{aligned}
&H(R_{\boldsymbol{\Phi_3}} | R_{\boldsymbol{\Phi_1}} = r_{\boldsymbol{\Phi_1}}, R_{\boldsymbol{\Phi_2}} = r_{\boldsymbol{\Phi_2}}) \\
&= -\sum_{r_{\boldsymbol{\Phi_3}} \in \{0,1\}} P(R_{\boldsymbol{\Phi_3}} = r_{\boldsymbol{\Phi_3}} | R_{\boldsymbol{\Phi_1}} = r_{\boldsymbol{\Phi_1}}, R_{\boldsymbol{\Phi_2}} = r_{\boldsymbol{\Phi_2}}) \cdot \\
&\qquad \log_2(P(R_{\boldsymbol{\Phi_3}} = r_{\boldsymbol{\Phi_3}} | R_{\boldsymbol{\Phi_1}} = r_{\boldsymbol{\Phi_1}}, R_{\boldsymbol{\Phi_2}} = r_{\boldsymbol{\Phi_2}})).
\end{aligned}$$

Figure 4 shows the Min and Shannon entropy for different ρ_{ij} values for an arbiter PUF with $n = 32$. Again, we see that the remaining APUF entropies depend on the correlations between the two known challenges and the other challenges. Once again, the min entropy is more dramatically reduced than the Shannon entropy, as expected. This means that there is an increase in the probability of guessing a third challenge correctly when two challenges are known. On the diagonal (yellow) the two known challenges do not reveal any information about the remaining challenges. While one CRP might give us some information about the unknown challenge, the other CRP leads us on the opposite direction, so we do not learn anything and the probability to guess the unknown challenge response is still 0.5, i.e. has entropy 1 bit.

C. Neighborhoods

We now ask how we may generalize the notion of similarity bins introduced earlier, to predictability and entropy neighborhoods. The question is which challenges $\mathbf{\Phi_3}$ lie within a certain "distance" / "semi-metric" d to two anchor challenges rather than one. We would like an algorithm to enumerate the challenges that lie within this predictability or entropy neighborhood, denoted by $B(d, \mathbf{\Phi_1}, \mathbf{\Phi_2}, r_1, r_2)$, and as a by-product, its size. To this end, define

$$\begin{aligned} &B(A, \mathbf{\Phi_1}, \mathbf{\Phi_2}, r_1, r_2) = \big\{\mathbf{\Phi_3} : \\ &\quad \max(P(R_{\mathbf{\Phi_3}} = 1 | R_{\mathbf{\Phi_1}} = r_1, R_{\mathbf{\Phi_2}} = r_2), \\ &\quad 1 - P(R_{\mathbf{\Phi_3}} = 1 | R_{\mathbf{\Phi_1}} = r_1, R_{\mathbf{\Phi_2}} = r_2)) = A\big\} \\ &B(H, \mathbf{\Phi_1}, \mathbf{\Phi_2}, r_1, r_2) = \big\{\mathbf{\Phi_3} : \\ &\quad H(R_{\mathbf{\Phi_3}} | R_{\mathbf{\Phi_1}} = r_{\mathbf{\Phi_1}}, R_{\mathbf{\Phi_2}} = r_{\mathbf{\Phi_2}}) = H\big\} \end{aligned}$$

In this "know 2, predict a third" case, we will see that not all prediction accuracies/entropies are possible. It depends on the relationship between the challenges you know. If you know two challenges, they are fixed, and hence so is their correlation ρ_{12}. We are thus interested in seeing how many challenges lie at different correlations ρ_{13}, ρ_{23} to these challenges.

In order to define the similarity neighborhoods and count them, we will need to quantify which correlation triples $(\rho_{12}, \rho_{13}, \rho_{23})$ are possible.

This is illustrated in Figure 5. To approach this, define

$$\begin{aligned} K :=&\ \text{number of indices in } S_{12} \text{ that we "keep" in } \mathbf{\Phi_3} \\ \overset{(a)}{=}&\ \frac{|S_{13}| + |S_{23}| + |S_{12}| - n}{2} \end{aligned} \tag{9}$$

where (a) follows by setting $x := |S_{13}| - K$ (the number of bits from D_{12} that belongs to $|S_{13}|$), and $y := |S_{23}| - K$ (the number of bits from D_{12} that belongs to $|S_{23}|$), from which, since x and y must cover the whole D_{12} set we see that $x + y = |D_{12}| = n - |S_{12}|$ and by replacing x and y, we obtain $K = \frac{|S_{13}|+|S_{23}|+|S_{12}|-n}{2}$.

1) Neighborhood construction: How can we find one challenge in $B(d, \mathbf{\Phi_1}, \mathbf{\Phi_2}, r_1, r_2)$? We enumerate the steps below which takes known challenges $\mathbf{\Phi_1}, \mathbf{\Phi_2}$ with correlation coefficient ρ_{12} and produces a single output challenge $\mathbf{\Phi_3}$ at the desired distance d in a chosen "semi-metric" space (accuracy, Shannon or min entropies).

This distance d depends on the challenge response values r_1 and r_2, and how to pick ρ_{13}, ρ_{23} to satisfy this equation:

1) Find ρ_{13}, ρ_{23} based on desired distance d and the set of all possible $(\rho_{12}, \rho_{13}, \rho_{23})$ tuples, given by Figure for the given ρ_{12}. How this Figure is obtained is presented in the Question below "Which correlations ρ_{13} and ρ_{23} are possible given ρ_{12}". This essentially boils down to picking ρ_{13}, ρ_{23} of the desired accuracy (represented by color) in our numerical evaluations.
2) The $\rho_{12}, \rho_{13}, \rho_{23}$ determine the sizes $|S_{12}|, |S_{13}|, |S_{23}|$ of the sets S_{12}, S_{13}, S_{23}. From here we can use Algorithm 3 in Appendix A which gives the following steps :
3) From $|S_{12}|, |S_{13}|, |S_{23}|$ we find K as in (9). Select any K indices in S_{12} to keep fixed in $\mathbf{\Phi_3}$. The remaining indices in S_{12} in $\mathbf{\Phi_3}$ must be flipped.
4) For the remaining indices in D_{12}, pick $|S_{13}| - K$ of them the same as $\mathbf{\Phi_1}$ and flip the remaining ones with respect to $\mathbf{\Phi_1}$, yielding the desired ρ_{13} and ρ_{23}.

2) Example: constructing a $\mathbf{\Phi_3} \in B(d, \mathbf{\Phi_1}, \mathbf{\Phi_2}, r_1, r_2)$: Given challenges $\mathbf{\Phi_1}$ and $\mathbf{\Phi_2}$, to create a third vector with desired correlations (note that not all will be possible), the main idea is to know how many bits we have to keep/fix from the first two challenges and how many we have to flip. Consider $n = 8$ and $\mathbf{\Phi_1} := (+1, +1, +1, +1, +1, +1, +1, +1, +1)$, $\mathbf{\Phi_2} := (+1, +1, +1, +1, -1, -1, -1, -1, +1)$, then $\rho_{12} = 0$. Say we wish to predict a third challenge $\mathbf{\Phi_3}$ with probability of accuracy $A \approx 0.6$. We follow the steps of the algorithm:

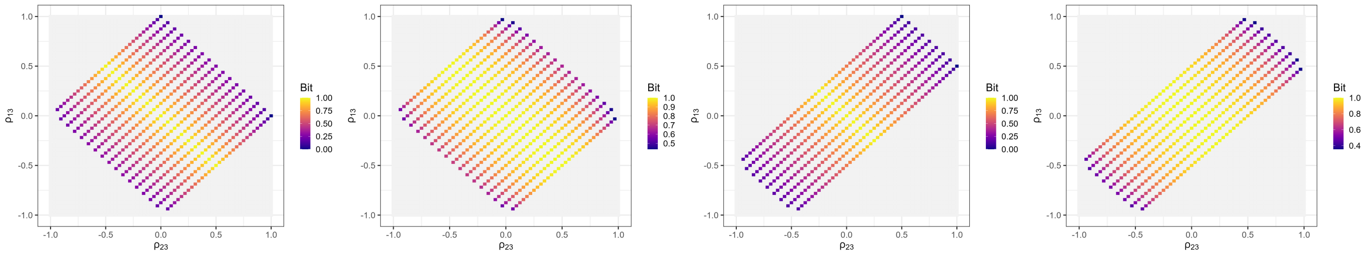
1) By generating the Figures from (12) – (13), we can see that $\rho_{23} = 0$, $\rho_{23} = 1/3$ and $\Phi_{1,1} = \Phi_{2,1} = \Phi_{3,1} = 1$ is one of the multiple possible choices.
2) This is equivalent to selecting $|S_{23}| = 4$ and $|S_{13}| = 6$.
3) To create a challenge with these desired correlations to the two known challenges, we need to decide how many indices in $\mathbf{\Phi_3}$ to keep from S_{12}, let us call this set of indices K, and then how many to keep and flip from the set D_{12}. Since $\Phi_{1,1} = \Phi_{2,1} = \Phi_{3,1} = 1$ the first bit belongs to the sets S_{12}, S_{13}, S_{23} so we have to select it in the ones we fix and $K - 1 = (|S_{12}| + |S_{13}| + |S_{23}| - n)/2 - 1 = (4 + 4 + 6 - 8)/2 - 1 = 2$ other bits to fix from the S_{12}. Take for example $\mathbf{\Phi_3} := (+1, *, +1, +1, *, *, *, *, +1)$ where the $*$ positions still need to be filled in.
4) Finally, we can flip the other bits in S_{12} to obtain $\mathbf{\Phi_3} := (+1, -1, +1, +1, *, *, *, *, +1)$. Then we select $|S_{13}| - K = 3$ bits from D_{12} to fix and flip the left ones as $\mathbf{\Phi_3} := (+1, -1, +1, +1, +1, +1, +1, -1, +1)$. Equivalently, we could have fixed $|S_{23}| - K = 1$ bits from D_{12} and flipped the left ones. Then, this $\mathbf{\Phi_3}$ can be predicted with accuracy $P(R_{\mathbf{\Phi_3}} = 1 | R_{\mathbf{\Phi_1}} = 1, R_{\mathbf{\Phi_2}} = 1) = \frac{1}{2}\left[1 + \frac{\arcsin\frac{1}{3} + \arcsin 0}{\frac{\pi}{2} + \arcsin 0}\right] \approx 0.61$.

The algorithm is given in Algorithm 3 in Appendix A.

3) Neighborhood size: So far we have produced one challenge of a given desired distance to the anchors. We now answer how many challenges exist with that ditance, and how can they be efficiently enumerated? The answer to this is derived directly from the way we created the single challenge – i.e. by looking at how many arbitrary choices we had. For example, in the case $\phi_{1,1} = \phi_{2,1} = \phi_{3,1}$: how many ways are there to choose $K - 1$ bits from $S_{12} - 1$ (since the first bit is fixed in this example) and $|S_{13}| - K$ bits from D_{12}, with $|D_{12}| = n - |S_{12}|$. Now, there are two possibilities, when $\phi_{1,1} = \phi_{2,1}$ and when $\phi_{1,1} \neq \phi_{2,1}$ which must be treated differently as the 1st position is special in the Φ notation.

- If $\phi_{1,1} = \phi_{2,1}$ and $|S_{13}| + |S_{23}| + |S_{12}| - n \equiv 0 \mod 2$, then do not flip first bit:

$$\begin{aligned} \#\text{of challenges } &= \binom{|S_{12}| - 1}{K - 1}\binom{n - |S_{12}|}{|S_{13}| - K} \\ &= \binom{|S_{12}| - 1}{K - 1}\binom{n - |S_{12}|}{|S_{23}| - K} \end{aligned} \tag{10}$$



(a) Min-Entropy : $r_1 = r_2$ and $\rho_{12} = 0$ (b) S-Entropy : $r_1 = r_2$ and $\rho_{12} = 0$ (c) Min-Entropy : $r_1 = r_2$ and $\rho_{12} = 0.5$ (d) S-Entropy : $r_1 = r_2$ and $\rho_{12} = 0.5$

Fig. 4: Min-entropy and Shannon entropy when knowing 2 challenges

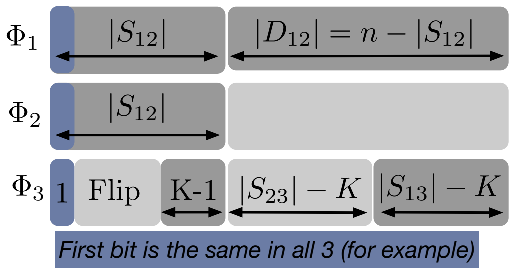


Fig. 5: Illustration of the sizes of the difference index sets $S_{12}, S_{13}, D_{12}, S_{23}$ and integer K representing the number of indices in S_{12} that we keep the same in $\mathbf{\Phi_3}$. In this example we assume the first bit is the same in all three $\mathbf{\Phi_{1,2,3}}$.

- If $\phi_{1,1} = \phi_{2,1}$ and $|S_{13}| + |S_{23}| + |S_{12}| - n \equiv 1 \mod 2$, then flip the first bit:

$$\begin{aligned}\#\text{of challenges} &= \binom{|S_{12}|-1}{K}\binom{n-|S_{12}|}{|S_{13}|-K} \\ &= \binom{|S_{12}|-1}{K}\binom{n-|S_{12}|}{|S_{23}|-K}\end{aligned}$$

- If $\phi_{1,1} \neq \phi_{2,1}$ we need $|S_{13}|+|S_{23}|+|S_{12}|-n \equiv 0 \mod 2$ since we necessarily have $\phi_{3,1} \neq \phi_{2,1}$ or $\phi_{3,1} \neq \phi_{1,1}$

$$\begin{aligned}&\#\text{of challenges} \qquad (11)\\ &= \binom{|S_{12}|}{K}\binom{n-|S_{12}|}{|S_{13}|-K-1} \text{ if } S(\mathbf{\Phi_1}, \mathbf{\Phi_3}) \equiv 0 \mod 1 \\ &= \binom{|S_{12}|}{K}\binom{n-|S_{12}|}{|S_{23}|-K-1} \text{ if } S(\mathbf{\Phi_2}, \mathbf{\Phi_3}) \equiv 0 \mod 1\end{aligned}$$

Notice that, contrary to the accuracy, the number of challenges does not depend on the challenge responses. To create similarity bins of size m with two anchors we can use Algorithm 2, which in turn calls Algorithms 4 and 5.

4) Neighborhood landscape discussion: Now we provide some in-depth analysis on the details of a neighborhood's landscape with the following questions.

a) Which correlations ρ_{13} and ρ_{23} are possible given ρ_{12}?: The answer depends on whether $\phi_{1,1}, \phi_{2,1}, \phi_{3,1}$ are the same or different. This is because this first bit has a different variance than the others, as per Lemma II.1. We present one case as an example; the others follow similarly.

Assume $\phi_1^1 = \phi_2^1 = \phi_3^1$. Then equation (9) and (10) imply two sufficient and necessary conditions on the correlation between the three challenges to be able to create the third one. In particular, by requiring $K \in \mathbb{N}$ in (9), and $\#\text{of challenges} = \binom{|S_{12}|-1}{K-1}\binom{n-|S_{12}|}{|S_{13}|-K} = \binom{|S_{12}|-1}{K-1}\binom{n-|S_{12}|}{|S_{23}|-K} > 0$ we obtain the following conditions:

$$\begin{aligned}&K \in \mathbb{N} \rightarrow |S_{13}| + |S_{23}| + |S_{12}| - n \equiv 0 \mod 2\\ &\binom{|S_{12}|-1}{K-1}\binom{n-|S_{12}|}{|S_{13}|-K} > 0 \rightarrow |S_{12}| \geq K\\ &\qquad\qquad \rightarrow n - |S_{12}| \geq |S_{13}| - K\\ &\binom{|S_{12}|-1}{K-1}\binom{n-|S_{12}|}{|S_{23}|-K} > 0 \rightarrow n - |S_{12}| \geq |S_{23}| - K\end{aligned}$$

Re-writing these using correlation coefficients (recall $|S| = (\rho+1)\frac{n}{2}$) yields the inequalities relating the possible correlations $\rho_{12}, \rho_{13}, \rho_{23}$ (which also must all lie in $[-1, 1]$):

$$\begin{aligned}\rho_{13} &\leq -\rho_{23} + 1 - \rho_{12} \qquad (12)\\ \rho_{13} &\geq \rho_{23} - 1 + \rho_{12}\end{aligned}$$

$$\begin{aligned}\rho_{13} &\leq \rho_{23} + 1 - \rho_{12}\\ \rho_{13} &\geq -\rho_{23} - 1 + \rho_{12} \qquad (13)\end{aligned}$$

One can visualize these equations easily, yielding rotated rectangles in the ρ_{13}, ρ_{23} plane for each given ρ_{12} (fixed by the two known challenges $\mathbf{\Phi_1}$ and $\mathbf{\Phi_2}$. The shapes in Figures 6 indicate which triples are possible, as shown by the linear equations above. The largest range of possible correlations occurs when $\rho_{12} = 0$, i.e. the first two challenges are uncorrelated. There are many such challenges, exactly how many is given by for example Figure 6.

b) How many challenges lie at different possible (ρ_{13}, ρ_{23}) from given challenges with correlation ρ_{12}?: To obtain this, we simply evaluate Equations (10) – (11) depending on what case we are in for each possible $(\rho_{12}, \rho_{13}, \rho_{23})$ triple. These yield the different colors in the yellow/blue plots of Figures 6. Notice that the densities do not depend on the actual values r_1, r_2 that the challenges take on.

c) What prediction accuracies and entropies are possible with different $(\rho_{12}, \rho_{13}, \rho_{23})$ triples?: Finally, given that we now know how many challenges there are at different correlations to one another, the question is how many challenges there are at different prediction accuracies or entropies. This is given by Theorem V.1 and shown in the purplish plots of Figures 4 and 6.

d) Numerical evaluations and interpretations of plots: Algorithms 2–5 in Appendix A allow us to create the bins $B(d, \mathbf{\Phi_1}, \mathbf{\Phi_2}, r_1, r_2)$ for all possible distance d in the "semi-metric" space defined by the accuracy or the entropies, and then the Figures 4 and 6. We now present some numerical evaluations to provide an understanding of the similarity bins when we know 2 challenges and wish to understand how many challenges lie at different distances to these two. To do so, we illustrate the following three questions:

1) Which correlations ρ_{13} and ρ_{23} are possible given ρ_{12}? This is given by the shape in Figures 6–**??** which depict equations (12) – (13).
2) What prediction distances are possible with different $(\rho_{12}, \rho_{13}, \rho_{23})$ triples? This is given by Theorem V.1 and is given by the purplish plots of Figures 4,6 and **??**.. The accuracy and entropies do depend on the actual values of the responses as well as the correlations. This may be intuitively thought of as follows: if the two challenges we know are well "aligned", i.e. when $r_1 \cdot r_2 \cdot \text{sign}(\rho_{12}) = +1$ this means the challenges have the same response and are highly correlated then they act more like one challenge with respect to the third one. Otherwise the information provided by the two challenges when ρ_{12} negative, $r_1 = r_2$ is somewhat contradictory, then information is neutralized for $\rho_{13} = \rho_{23}$. The "blue line" on the accuracy graphs shows where the information is neutralised: line $y = -x$ if $r_1 = r_2$, $y = x$ if $r_1 \neq r_2$.
3) How many challenges lie at different possible (ρ_{13}, ρ_{23}) from given challenges with correlation ρ_{12}? This is given by equations (10) – (11) and is given by the blue/yellow plots of Figures 6.

e) Expected conditional entropy and accuracy of a challenge's response: Knowing the values of the entropies and accuracy in each neighborhood as well as the number of challenges in all of them. we can compute the expected conditional entropy and accuracy of a challenge's response. Calling d_i the distance in $B(d_i, \mathbf{\Phi_1}, \mathbf{\Phi_2}, r_1, r_2)$ in the entropy or accuracy space and n^2 the maximum number of different neighborhoods knowing that if one is not possible then $|B(d_i, \mathbf{\Phi_1}, \mathbf{\Phi_2}, r_1, r_2)| = 0$. The expected semi-metric $\bar{d}$ is then given as follows:

$$\bar{d} = \frac{1}{\#\text{possible challenges}} \sum_{i=1}^{n^2} d_i |B(d_i, \mathbf{\Phi_1}, \mathbf{\Phi_2}, r_1, r_2)|.$$

Table II shows the different expected conditional entropy and accuracy of a challenge's response in function of the correlation ρ_{12} between the two anchors and the number of stages of the PUF, n.

There is quite a bit of information packed into these Figures. Some things to note include: highly predictable challenges have the yellow color in the left (a) plots: they tend to be around the edges and there are relatively few of them. Less predictable challenges are found along the dark blue/purplish lines in the left (a) plots and there tend to be many of them. However, sometimes we can see large clumps (density plots are greenish) that are also reasonably well predicted (pinkish). Overall, the hope is that such plots will be useful to PUF protocol design engineers to give them an idea of the landscape of challenges: how many there are at different prediction accuracies or conditional response entropies once one or two challenges are exposed.

VI. Conclusion

We have presented two new tools for understanding and exploiting the CRP correlations of APUF (and by almost immediate extension, compute then entropy). The first is the response similarity, or the probability that two challenges yield the same response, which is a function of the challenges themselves. This may be used to optimally predict the response to one challenge given the response to another challenge, and to obtain the conditional response entropy, The second is the derivation of a simple algorithm for enumerating the similarity bins, or entropy bins to a given challenge, i.e. all challenges that have the same response similarity or conditional response entropy to a given anchor challenge. In combination, these form a powerful tool for understanding the CRP landscape: how CRPs are correlated and may be used to obtain statistical properties of linear-threshold-based PUFs.

Declarations

- *Ethical Approval:* This work is original and has not been published elsewhere, nor is it currently under consideration for publication elsewhere.
- *Competing interests:* The authors have no relevant financial or non-financial interests to disclose.
- *Authors' contributions:* All authors contributed to the study conception and design. Material preparation, simulations and analysis were performed by Vincent Dumoulin, Natasha Devroye and Wenjing Rao. The first draft of the manuscript was written by Vincent Dumoulin and Natasha Devroye then all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.
- *Funding:* This work was supported by the National Science Foundation under award 2217023 and 2244479.
- *Data:* No exterior data set is used. All simulations and plots are derived based on the statistical models, equations and pseudo algorithms presented in the paper.

References

[1] W. Che, F. Saqib, and J. Plusquellic, "Puf-based authentication," in *2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov 2015, pp. 337–344.

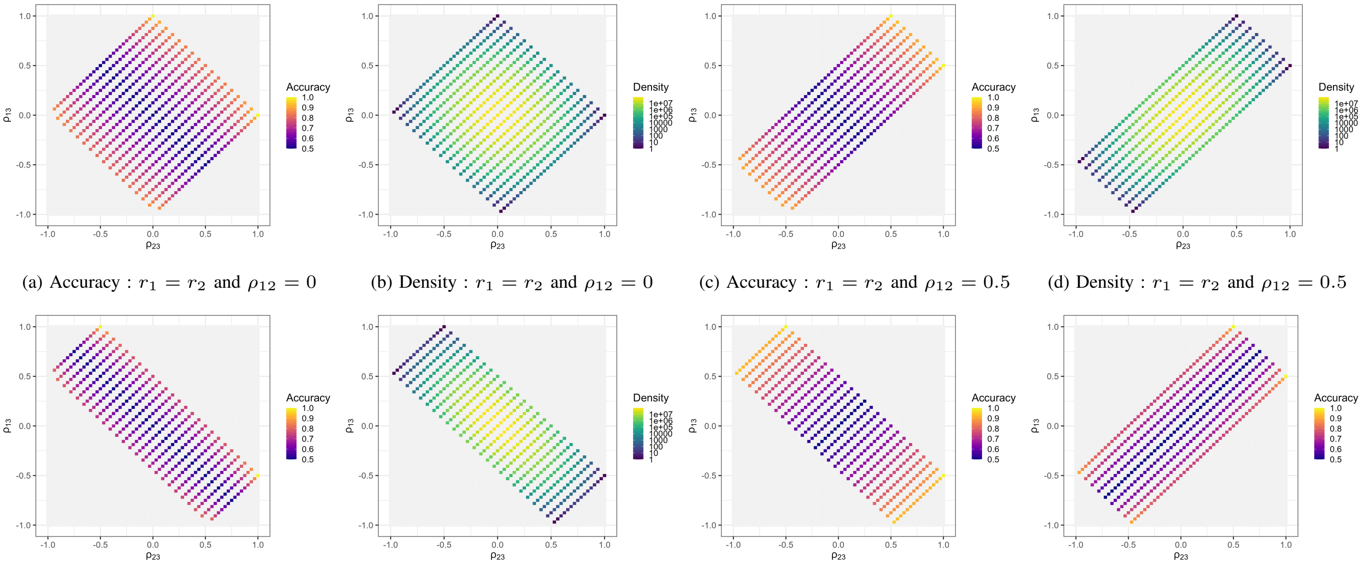
[2] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," *9th ACM Conference on Computer and Communication Security*, 2002.

[3] L. Feiten, M. Sauer, and B. Becker, "On metrics to quantify the inter-device uniqueness of PUFs," Cryptology ePrint Archive, Paper 2016/320, 2016. [Online]. Available: https://eprint.iacr.org/2016/320

[4] Y. Hori, T. Yoshida, T. Katashita, and A. Satoh, "Quantitative and statistical performance evaluation of arbiter physical unclonable functions on fpgas," *Int. Conf. on Reconfigurable Computing and FPGAs*, pp. 298–303, 2010.

[5] Rukhin, A. et al., "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications," 2010.

[6] M. Majzoobi, M. Rostami, F. Koushanfar, D. S. Wallach, and S. Devadas, "Slender puf protocol: A lightweight, robust, and secure authentication by substring matching," in *Security and Privacy Workshops (SPW), 2012 IEEE Symposium on.* IEEE, 2012, pp. 33–44.



(a) Accuracy : $r_1 = r_2$ and $\rho_{12} = 0$ (b) Density : $r_1 = r_2$ and $\rho_{12} = 0$ (c) Accuracy : $r_1 = r_2$ and $\rho_{12} = 0.5$ (d) Density : $r_1 = r_2$ and $\rho_{12} = 0.5$

(e) Accuracy : $r_1 = r_2$ and $\rho_{12} = -0.5$ (f) Density : $r_1 = r_2$ and $\rho_{12} = -0.5$ (g) Accuracy : $r_1 \neq r_2$ and $\rho_{12} = -0.5$ (h) Accuracy : $r_1 \neq r_2$ and $\rho_{12} = 0.5$

Fig. 6: Accuracy and density plot same and different responses.

	n=32			**n=64**			**n=128**		
	$\rho_{12} = -0.5$	$\rho_{12} = 0$	$\rho_{12} = 0.5$	$\rho_{12} = -0.5$	$\rho_{12} = 0$	$\rho_{12} = 0.5$	$\rho_{12} = -0.5$	$\rho_{12} = 0$	$\rho_{12} = 0.5$
H-min	0.8259	0.8329	0.8450	0.8728	0.8788	0.8879	0.9080	0.9127	0.9194
H-Shannon	0.9798	0.9815	0.9842	0.9898	0.9908	0.9922	0.9948	0.9954	0.9961
Accuracy	0.5663	0.5634	0.5584	0.5472	0.5448	0.5413	0.5335	0.5317	0.5292

TABLE II: Expected conditional entropies and accuracies of a challenge's response knowing two CRPs

[7] M. Rostami, M. Majzoobi, F. Koushanfar, D. S. Wallach, and S. Devadas, "Robust and reverse-engineering resilient puf authentication and key-exchange by substring matching," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 1, pp. 37–49, 2014.

[8] Y. M.-D. et al., "A noise bifurcation architecture for linear additive physical functions," in *2014 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*. IEEE, 2014, pp. 124–129.

[9] J. Delvaux, R. Peeters, D. Gu, and I. Verbauwhede, "A survey on lightweight entity authentication with strong pufs," *ACM Comput. Surv.*, vol. 48, no. 2, Oct. 2015. [Online]. Available: https://doi.org/10.1145/2818186

[10] M. Majzoobi, F. Koushanfar, and M. Potkonjak, "Lightweight secure pufs," in *2008 IEEE/ACM International Conference on Computer-Aided Design*, Nov 2008, pp. 670–673.

[11] ——, "Testing techniques for hardware security," in *2008 IEEE International Test Conference*, Oct 2008, pp. 1–10.

[12] J. Delvaux, "Machine-learning attacks on PolyPUFs, OB-PUFs, RPUFs, LHS-PUFs, and PUFĐFSMs," *IEEE Transactions on Information Forensics and Security*, pp. 1–1, 2019.

[13] P. H. Nguyen, D. P. Sahoo, R. S. Chakraborty, and D. Mukhopadhyay, "Security analysis of arbiter puf and its lightweight compositions under predictability test," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 22, no. 2, pp. 20:1–20:28, Dec. 2016. [Online]. Available: http://doi.acm.org/10.1145/2940326

[14] O. Rioul, P. Solé, S. Guilley, and J.-L. Danger, "On the entropy of physically unclonable functions," in *ProcISIT*, July 2016, pp. 2923–2932.

[15] M.-D. Yu, R. Sowell, A. Singh, D. M'Raïhi, and S. Devadas, "Performance metrics and empirical results of a puf cryptographic key generation asic," in *HOST*. IEEE, 2012, pp. 108–115.

[16] A. Schaub, O. Rioul, and J. J. Boutros, "Entropy estimation of physically unclonable functions via chow parameters," *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 698–704, 2019.

[17] Frisch, C., Wilde, F., Holzner, T. et al., "A practical approach to estimate the min-entropy in pufs," *J Hardw Syst Secur*, vol. 7, pp. 138–146, 2023.

[18] Jao, J., Wilcox, I., Thotakura, S. et al.., "An Analysis of FPGA LUT Bias and Entropy for Physical Unclonable Functions," *J Hardw Syst Secur*, vol. 7, p. 110–123, 2023.

[19] Y. Lao and K. K. Parhi, "Statistical analysis of mux-based physical unclonable functions," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 5, pp. 649–662, May 2014.

[20] W. Che, V. K. Kajuluri, M. Martin, F. Saqib, and J. Plusquellic, "Analysis of entropy in a hardware-embedded delay puf," in *Cryptography*, Jun. 2017.

[21] W. Stefani, F. Kappelhoff, M. Gruber, Y.-N. Wang, S. Achour, D. Mukhopadhyay, and U. Rührmair, "Strong PUF security metrics: Sensitivity of responses to single challenge bit flips," *Cryptology ePrint Archive, Paper 2024/378*, 2024.

[22] D. Boning and S. Nassif, "Models of process variations in device and interconnect," *Design of high performance microprocessor circuits*, p. 6, 2000.

[23] J. Tobisch and G. T. Becker, "On the scaling of machine learning attacks on pufs with application to noise bifurcation," in *Revised Selected Papers of the 11th International Workshop on Radio Frequency Identification - Volume 9440*, ser. RFIDsec 2015. New York, NY, USA: Springer-Verlag New York, Inc., 2015, pp. 17–31.

[24] W. Yeqi, F. Tim, D. Vincent, R. Wenjing, and D. Natasha, "Apuf faults: Impact, testing, and diagnosis," pp. 442–447, 2022.

[25] T. Cover and J. Thomas, *Elements of Information Theory*, 2nd ed. New York:Wiley, 2006.

[26] M. C. Cheng, "The orthant probabilities of four gaussian variates," *The Annals of Mathematical Statistics*, vol. 40, no. 1, pp. 152–161, 1969. [Online]. Available: http://www.jstor.org/stable/2239206

[27] J. Massey, "Guessing and entropy," in *ProcISIT*, Jul 1995, p. 204.

Appendix

Algorithms for bin creation

Algorithm 1 builds a similarity bin. For a different "semimetric", use the theoretical results or the graph in Figure 2 to get the two $\mathbf{s} = (s_1, s_2)$ factors that will give you the desired value/ create a similarity bin for each s_i (two similarity factors for entropy and accuracy). Form the union of the two disjoint bins.

Algorithm 1: Bin construction for one anchor:

```
Input: anchor $\mathbf{\Phi}, n$, similarity factor(s) $\mathbf{s} = (s_1, s_2)$
       derived from a wanted "semimetric" (entropy,
       accuracy,probability).
Output: Bin $B(\mathbf{s}, \mathbf{\Phi})$.
// initialize
$B \leftarrow \emptyset$
// use $s$ and $\phi_1$ to determine $\phi_1'$.
if $s$ mod $1! = 0$ then
    $\phi_1' = -\phi_1$ ;                        // $s$ even
else
    $\phi_1' = \phi_1$ ;                         // $s$ odd
/* use $s$ to decide how many of $\phi_2, \cdots, \phi_n$, to
   flip to construct $\phi_2', \cdots, \phi_n'$.              */
$F \leftarrow \{f | f \in \{0,1\}^{n-1}, \sum f_i = \lfloor s \rfloor\}$
/* $F$ contains all the strings of size $n-1$
   with Hamming weight $\lfloor s \rfloor$, and $f$ encodes where
   to flip $\phi_2, \cdots, \phi_n$ for $\phi_i'$'s           */
for $f \in F$ do
    for $i$ = 2 to $n$ do
        if $f_{i-1} == 0$ then
            $\phi_i' = \phi_i'$
        else
            $\phi_i' = -\phi_i'$
    $\mathbf{\Phi}' \leftarrow (\phi_1', \phi_2', \cdots, \phi_n', +1)$
    $B \leftarrow B \bigcup \{\mathbf{\Phi}'\}$
return $B$ // This is $B(\mathbf{s}, \mathbf{\Phi})$
```

Algorithm 2: Bin construction with two anchors

```
Input: Anchors $\mathbf{\Phi_1}, \mathbf{\Phi_2}$ and responses $r_1$ $r_2$
Output: Similarity bins of size $m$ for all possible
        distance $d : B(d, \mathbf{\Phi_1}, \mathbf{\Phi_2}, r_1, r_2)$
// initialize: look for the bits to select
for $|S_{12}| = 1$ to $n$ do
    for $|S_{13}| = 1$ to $n$ do
        for $|S_{23}| = 1$ to $n$ do
            $\mathbf{K} \leftarrow (|S_{13}| + |S_{23}| + |S_{12}| - n)/2$;
            if $\Phi_{1,1} = \Phi_{2,1}$ then
                CBSS($\mathbf{\Phi_1}, \mathbf{\Phi_2}$, $r_1$, $r_2$, $|S_{12}|$, $|S_{13}|$, $|S_{23}|$)
            if $\Phi_{1,1} \neq \Phi_{2,1}$ then
                CBDS($\mathbf{\Phi_1}, \mathbf{\Phi_2}$, $r_1$, $r_2$, $|S_{12}|$, $|S_{13}|$, $|S_{23}|$)
            else Ignore // Challenge does not exist
    return $B(d, \mathbf{\Phi_1}, \mathbf{\Phi_2}, r_1, r_2)$ for all distances $d$
```

Algorithm 3 builds a third challenge at a distance d in a the chosen "semimetric" space (accuracy or entropy space). Algorithms 2-5 create the bins using the previous algorithm.

Algorithm 3: CC: Create a 3rd Challenge $\mathbf{\Phi_3}$

```
Input: Anchors $\mathbf{\Phi_1}, \mathbf{\Phi_2}$, responses $r_1$ $r_2$,
       $|S_{12}|$, $|S_{13}|$, $|S_{23}|$, $\Phi_{3,1}$
Output: a challenge $\mathbf{\Phi_3}$ with a desired distance d
// construct $\Phi_3$ from $\rho_{12}, \rho_{13}, \rho_{23}$ we know
   $\Phi_{1,1} \overset{?}{=} \Phi_{2,1} \overset{?}{=} \Phi_{3,1}$
$\mathbf{K} \leftarrow (|S_{13}| + |S_{23}| + |S_{12}| - n)/2$;
if $\Phi_{1,1} = \Phi_{2,1} = \Phi_{3,1}$ then
    if $\mathbf{K} \in \mathbb{N}$ then
        Fix: the first bit and $\mathbf{K} - 1$ other bits in $S_{12}$;
        Flip: all other bits in $S_{12}$;
        Fix: any $|S_{13}| - K$ bits in $D_{12}$;
        Flip: all other bits in $D_{12}$ ;
        // equivalently fix $|S_{23}| - K$ bits in $D_{12}$,
           and flip the rest
    else
        return False;
        // # of bits to fix is not an integer
else if $\Phi_{1,1} = \Phi_{2,1} \neq \Phi_{3,1}$ then
    if $\mathbf{K} = x.5$ then
        Fix: any $\lfloor \mathbf{K} \rfloor$ bits except the 1st bit in $S_{12}$;
        Flip: all other in $S_{12}$ (including the 1st bit);
        Fix: any $|S_{13}| - K$ bits in $D_{12}$;
        Flip: all other bits in $D_{12}$;
        // equivalently fix $|S_{23}| - K$ bits in $D_{12}$,
           and flip the rest
    else
        return False;
else if $\Phi_{1,1} \neq \Phi_{2,1} = \Phi_{3,1}$ then
    if $\mathbf{K} \in \mathbb{N}$ then
        Use: the first bit of $\Phi_{2,1}$;
        Fix: any $\mathbf{K}$ bits in $S_{12}$;
        Flip: all other bits in $S_{12}$;
        Fix: any $|S_{23}| - K - 1$ bits in $D_{12}$;
        Flip: all other bits in $D_{12}$;
    else
        return False;
        // Number of bits to fix is not integer
else if $\Phi_{1,1} = \Phi_{3,1} \neq \Phi_{2,1}$ then
    if $\mathbf{K} \in \mathbb{N}$ then
        Use: the first bit of $\Phi_{3,1}$ ;
        Fix: any $\mathbf{K}$ bits in $S_{12}$ ;
        Flip: all other bits in $S_{12}$ ;
        Fix: any $|S_{13}| - K - 1$ bits in $D_{12}$ ;
        Flip: all other bits in $D_{12}$ ;
    else
        return False;
        // Number of bits to fix is not integer
return $\Phi_3$ // Challenge $\mathbf{\Phi_3}$ has desired distance
```

Algorithm 4: CBSS: Create Bin first bit Same Sign: $\Phi_{1,1} = \Phi_{2,1}$

```
Input: Anchors $\mathbf{\Phi_1}, \mathbf{\Phi_2}$, responses $r_1$ $r_2$,
       $|S_{12}|$, $|S_{13}|$, $|S_{23}|$ under $\Phi_{1,1} = \Phi_{2,1}$
Output: $B(d, \mathbf{\Phi_1}, \mathbf{\Phi_2}, r_1, r_2)$ of size $m$
$\rho_{12} = (2|S_{12}|)/n - 1$;
begin
    $\Phi_{3,1} = \Phi_{1,1}$;
    $\rho_{13} = (2|S_{13}|)/n - 1$;
    $\rho_{23} = (2|S_{23}|)/n - 1$;
    #of challenges $= \binom{|S_{12}|-1}{K-1}\binom{n-|S_{12}|}{|S_{13}|-K}$;
    if K $\in \mathbb{N}$ & #of challenges $\geq 0$ then
        $p \leftarrow \frac{1}{2}\left[1 + \frac{r_1 \arcsin\rho_{13} + r_2 \arcsin\rho_{23}}{\pi/2 + r_1 r_2 \arcsin\rho_{12}}\right]$;
        $d_1$ = Accuracy(p) or entropy(p);
        for $i = 1$ to $m$ do
            $\Phi_3 \leftarrow$
              CC($\mathbf{\Phi_1}, \mathbf{\Phi_2}$, $r_1$, $r_2$, $|S_{12}|$, $|S_{13}|$, $|S_{23}|$, $\Phi_{3,1}$);
            $B(d_1, \mathbf{\Phi_1}, \mathbf{\Phi_2}, r_1, r_2) \leftarrow$
              $B(d_1, \mathbf{\Phi_1}, \mathbf{\Phi_2}, r_1, r_2) \bigcup \{\Phi_3\}$
    else Ignore // Challenge does not exist
begin
    $\Phi_{3,1} = -\Phi_{1,1}$;
    $\rho_{13} = (2|S_{13}| + 1)/n - 1$;
    $\rho_{23} = (2|S_{23}| + 1)/n - 1$;
    #of challenges $= \binom{|S_{12}|-1}{K}\binom{n-|S_{12}|}{|S_{13}|-K}$;
    if K $\in \mathbb{N}$ & #of challenges $\geq 0$ then
        $p \leftarrow \frac{1}{2}\left[1 + \frac{r_1 \arcsin\rho_{13} + r_2 \arcsin\rho_{23}}{\pi/2 + r_1 r_2 \arcsin\rho_{12}}\right]$;
        $d_2$ = Accuracy(p) or entropy(p);
        for $i = 1$ to $m$ do
            $\Phi_3 \leftarrow$
              CC($\mathbf{\Phi_1}, \mathbf{\Phi_2}$, $r_1$, $r_2$, $|S_{12}|$, $|S_{13}|$, $|S_{23}|$, $\Phi_{3,1}$);
            $B(d_2, \mathbf{\Phi_1}, \mathbf{\Phi_2}, r_1, r_2) \leftarrow$
              $B(d_2, \mathbf{\Phi_1}, \mathbf{\Phi_2}, r_1, r_2) \bigcup \{\Phi_3\}$
    else Ignore // Challenge does not exist
return $B(d_1, \mathbf{\Phi_1}, \mathbf{\Phi_2}, r_1, r_2)$, $B(d_2, \mathbf{\Phi_1}, \mathbf{\Phi_2}, r_1, r_2)$
```

Algorithm 5: CBDS: Create Bin first bit Different Sign: $\Phi_{1,1} \neq \Phi_{2,1}$

```
Input: Anchors $\mathbf{\Phi_1}, \mathbf{\Phi_2}$, responses $r_1$ $r_2$,
       $|S_{12}|$, $|S_{13}|$, $|S_{23}|$ under $\Phi_{1,1} \neq \Phi_{2,1}$
Output: $B(d, \mathbf{\Phi_1}, \mathbf{\Phi_2}, r_1, r_2)$ of size $m$
$\rho_{12} = (2|S_{12}| + 1)/n - 1$;
begin
    $\Phi_{3,1} = \Phi_{1,1}$;
    $\rho_{13} = (2|S_{13}|)/n - 1$;
    $\rho_{23} = (2|S_{23}| + 1)/n - 1$;
    #of challenges $= \binom{|S_{12}|}{K}\binom{n-|S_{12}|}{|S_{13}|-K-1}$;
    if K $\in \mathbb{N}$ & #of challenges $\geq 0$ then
        $p \leftarrow \frac{1}{2}\left[1 + \frac{r_1 \arcsin\rho_{13} + r_2 \arcsin\rho_{23}}{\pi/2 + r_1 r_2 \arcsin\rho_{12}}\right]$;
        $d_1$ = Accuracy(p) or entropy(p);
        for $i = 1$ to $m$ do
            $\Phi_3 \leftarrow$
              CC($\mathbf{\Phi_1}, \mathbf{\Phi_2}$, $r_1$, $r_2$, $|S_{12}|$, $|S_{13}|$, $|S_{23}|$, $\Phi_{3,1}$);
            $B(d_1, \mathbf{\Phi_1}, \mathbf{\Phi_2}, r_1, r_2) \leftarrow$
              $B(d_1, \mathbf{\Phi_1}, \mathbf{\Phi_2}, r_1, r_2) \bigcup \{\Phi_3\}$
    else Ignore // Challenge does not exist
begin
    $\Phi_{3,1} = \Phi_{2,1}$;
    $\rho_{13} = (2|S_{13}| + 1)/n - 1$;
    $\rho_{23} = (2|S_{23}|)/n - 1$;
    #of challenges $= \binom{|S_{12}|}{K}\binom{n-|S_{12}|}{|S_{23}|-K-1}$;
    if K $\in \mathbb{N}$ & #of challenges $\geq 0$ then
        $p \leftarrow \frac{1}{2}\left[1 + \frac{r_1 \arcsin\rho_{13} + r_2 \arcsin\rho_{23}}{\pi/2 + r_1 r_2 \arcsin\rho_{12}}\right]$;
        $d_2$ = Accuracy(p) or entropy(p);
        for $i = 1$ to $m$ do
            $\Phi_3 \leftarrow$
              CC($\mathbf{\Phi_1}, \mathbf{\Phi_2}$, $r_1$, $r_2$, $|S_{12}|$, $|S_{13}|$, $|S_{23}|$, $\Phi_{3,1}$);
            $B(d_2, \mathbf{\Phi_1}, \mathbf{\Phi_2}, r_1, r_2) \leftarrow$
              $B(d_2, \mathbf{\Phi_1}, \mathbf{\Phi_2}, r_1, r_2) \bigcup \{\Phi_3\}$
    else Ignore // Challenge does not exist
return $B(d_1, \mathbf{\Phi_1}, \mathbf{\Phi_2}, r_1, r_2)$, $B(d_2, \mathbf{\Phi_1}, \mathbf{\Phi_2}, r_1, r_2)$
```