Upper Bounds on The Min-Entropy of RO Sum, Arbiter, Feed-Forward Arbiter, and S-ArbRO PUFs

Jeroen Delvaux
KU Leuven, ESAT/COSIC and iMinds, Belgium, and Shanghai Jiao Tong Univeristy, CSE/LoCCS, China
Email: jdelvaux@esat.kuleuven.be

Dawu Gu
Shanghai Jiao Tong University, CSE/LoCCS, China
Email: dwgu@sjtu.edu.cn

Ingrid Verbauwhede
KU Leuven, ESAT/COSIC and iMinds, Belgium
Email: iverbauw@esat.kuleuven.be

***Abstract*—The focus and novelty of this work is the derivation of tight upper bounds on the min-entropy of several *physically unclonable funcions* (PUFs), i.e., Ring Oscillator Sum, Arbiter, Feed-Forward Arbiter, and S-ArbRO PUFs. This constrains their usability for the *fuzzy extraction* of a secret key, as an alternative to storing keys in non-volatile memory. For example, it is shown that an ideal Arbiter PUF with 64 stages cannot provide more than 197 bits of min-entropy. At Financial Cryptography 2012, Van Herrewege et al. assume that 1785 bits of min-entropy can be extracted, which renders their 128-bit key generator instantly insecure. We also derive upper bounds that comply with non-ideal PUFs, attributed to, e.g., manufacturing in silicon. As a side contribution hereby, we refute the claim that S-ArbRO PUFs are highly resistant against machine learning.**

I. Introduction

Physically unclonable functions (PUFs) are electrical circuits that are designed to be prone to manufacturing variations while remaining fairly robust against noise. From a black-box perspective, a binary input, i.e., the *challenge*, is mapped to a binary device-specific and slightly noisy output, i.e., the *response*. For so-called *strong PUFs*, the number of challenges scales exponentially with the circuit area. Although the latter type of PUF is most frequently used within an entity authentication protocol [1], key generation comprehends another application of interest.

Unfortunately, concatenating the responses to a list of publicly known challenges does not immediately result in a device-unique secret key. Apart from the noisiness, non-uniformities are bound to occur. Most notably, challenge-response pairs (CRPs) are far from independent. One manifestation thereof is that machine learning algorithms allow for the construction of an accurate predictive model, given a relatively small training set of CRPs [2]. A fuzzy extractor [3] nevertheless provides an *information-theoretically secure* mechanism to transform the response bits into a stable secret key. However, to guarantee that the key is indeed uniformly distributed, a tight lower bound on the ingoing min-entropy needs to be evaluated, which might very well be infeasible for strong PUFs [4].

Deriving tight upper bounds on the min-entropy of a strong PUF is a more feasible alternative. Although not usable for the secure instantiation of a key generator, it nevertheless results in valuable insights. To the best of our knowledge, feeding response bits into a lossless compression algorithm is the only tool available so far. The average number of retained bits comprehends an upper bound on the min-entropy. However, challenges and therefore also functional correlations among CRPs are not taken into account. The produced upper bound is hence not so very tight. Katzenbeisser et al. [5] nevertheless observe a considerable compression rate for the 65nm *arbiter PUFs* that were manufactured in the European research project UNIQUE. This can mainly be attributed to bias though, i.e., an imbalance in the number of zeros and ones, and hence not to functional correlations.

A. Contribution

We are the first to derive tight upper bounds on the min-entropy of various strong PUFs, hereby incorporating the highly correlated functional behavior. Although techniques are not necessarily limited thereto, we focus on Ring Oscillator (RO) Sum [6], S-ArbRO-2 [7], S-ArbRO-4 [7], Arbiter [8], and Arbiter Feed-Forward [8] PUFs. Two newly developed methods produce bounds via a black-box and white-box approach respectively, i.e., the internal mechanisms of the PUF are disregarded and explicitly incorporated respectively. Both approaches demonstrate that designers are often too optimistic about the min-entropy provided. The key generator implemented by Van Herrewege et al. [9] does not meet its intended 128-bit security level, for instance.

The black-box approach relies on easy-to-obtain machine learning results in order to evaluate the bound. PUFs simulated in software as well as hardware implementations on either a *field-programmable gate array* (FPGA) or an *application-specific integrated circuit* (ASIC) can be modeled and are therefore supported. The more effective the learning algorithm, the tighter the bound on the min-entropy. As a side contribution of independent interest, we refute the claim that S-ArbRO PUFs are highly resistant against machine learning.

The white-box approach requires mathematical analysis of the PUF internals. We rely on a commonly used variability model in order to render this approach feasible. Although hardware implementations do not necessarily comply with the latter assumptions, it still reflects the ideal-case behavior of the PUF as intended by their designers. Compared to the

black-box approach, bounds on the min-entropy are further improved.

B. Organization

Section II introduces notation and preliminaries. Section III and IV derive the black-box and white-box bounds respectively. Section V concludes the work.

II. Preliminaries

A. Notation

Vectors are denoted by a bold lowercase character, e.g., $\boldsymbol{x}$. All vectors are column vectors. Matrices are denoted by a bold uppercase character, e.g., $\boldsymbol{B}$. A random variable is denoted by an uppercase character, e.g., X. Its corresponding set of outcomes is denoted by an uppercase calligraphic character, e.g., $\mathcal{X}$. Consider the standard normal distribution $N(0,1)$, having zero mean and standard deviation $\sigma = 1$. Its *probability density function* and *cumulative distribution function* are denoted by $\mathsf{f}_{\mathsf{norm}}(\cdot)$ and $\mathsf{F}_{\mathsf{norm}}(\cdot)$ respectively. Probabilities and expected values are denoted by $\mathbb{P}(\cdot)$ and $\mathbb{E}[\cdot]$ respectively.

B. Min-Entropy Definitions

The *min-entropy* of a binary random variable X as defined in (1) is a worst-case predictability measure. Consider now a pair of possibly correlated binary random variables: (X, P). The *conditional min-entropy* [3] of X given P is as defined in (2). Terms of the expectation with $\mathbb{P}(P = \boldsymbol{p}) = 0$ are evaluated as 0. Both definitions quantify the probability that an attacker guesses a secret $\boldsymbol{x} \leftarrow X$ first time right, on a logarithmic scale.

$$\mathbb{H}_\infty(X) = -\log_2\big(\max_{\boldsymbol{x}\in\mathcal{X}} \mathbb{P}(X = \boldsymbol{x})\big). \tag{1}$$

$$\widetilde{\mathbb{H}}_\infty(X|P) = -\log_2\Big(\mathbb{E}_{\boldsymbol{p}\leftarrow P}\big[\max_{\boldsymbol{x}\in\mathcal{X}} \mathbb{P}((X = \boldsymbol{x})|(P = \boldsymbol{p}))\big]\Big). \tag{2}$$

A relevant issue in the design of cryptographic systems is that the min-entropy of a random variable X with an unknown distribution cannot be determined in an exhaustive experimental manner. The length of, e.g., secret keys and concatenated PUF responses, ranges from hundreds to thousands of bits in order to render brute-force attacks computationally infeasible. Therefore, one cannot simply measure the probability of occurrence of the most outcome of X. For silicon PUFs in particular, one would have to manufacture and read-out an infeasible number of devices for this purpose.

C. Delay-Based Strong PUFs

We specify five strong PUF designs that rely on the propagation delay of logic gates, or better, the variability thereof. Formally, we specify a function $r \leftarrow \mathsf{PUF}(\boldsymbol{c}, \boldsymbol{v})$, with r a single response bit, $\boldsymbol{c}$ the binary challenge, and $\boldsymbol{v}$ aggregating the relevant delay-based quantities. For convenience, $\boldsymbol{\gamma}$ denotes an invertible transformation of challenge $\boldsymbol{c}$.

The first three PUF designs rely on an array of ring oscillators (ROs). These are self-oscillating loops that consist of an odd number of inverters, i.e., NOT gates. The frequency of oscillation f, i.e., the reciprocal of the total propagation delay, depends on manufacturing variability. Various experimental work shows that a normal distribution $F \sim N(\mu_f, \sigma_f)$ captures reality quite accurately. Consider the measurement of m pairwise frequency differences: $\boldsymbol{v} = \begin{pmatrix} f_2 - f_1 & f_4 - f_3 & \dots & f_{2m} - f_{2m-1} \end{pmatrix}^T$.

A first design is the RO Sum PUF of Yu and Devadas [6]. Challenge $\boldsymbol{c}$ determines for each pair $i \in [1, m]$ whether either v_i or its opposite value $-v_i$ is accumulated to a variable Δf. Written as a dot product, $\Delta f = \boldsymbol{v}^T \cdot \boldsymbol{\gamma}$, with $\gamma_i \in \{-1, 1\}$. Thresholding $\Delta f \lessgtr 0$ results in a single response bit r. The number of challenges, i.e., $|\mathcal{C}| = 2^m$, scales exponentially with the circuit area.

Similarly, S-ArbRO-2 PUFs were proposed by Ganta and Nazhandali [7]. This design generalizes an RO Sum PUF so that only a subset of k oscillator pairs contributes to Δf, with $k \in [1, m]$ a fixed parameter. The selection of k pairs is part of the challenge $\boldsymbol{c}$. Thresholding of $\Delta f = \boldsymbol{v}^T \cdot \boldsymbol{\gamma}$ is subject to constraints $\gamma_i \in \{-1, 0, 1\}$ and $\sum_{i=1}^m |\gamma_i| = k$. The number of challenges $|\mathcal{C}|$ equals $\binom{m}{k} 2^k$.

The same authors also proposed S-ArbRO-4 PUFs. The set of m RO pairs, with m even, is partitioned into $m/2$ clusters with two pairs each. Only a subset of k clusters contributes to Δf, with $k \in [1, m/2]$ a fixed parameter. The selection of k clusters is again part of the challenge $\boldsymbol{c}$. Within each contributing cluster, either one out of two frequency differences is accumulated to Δf, without flipping signs. Thresholding of $\Delta f = \boldsymbol{v}^T \cdot \boldsymbol{\gamma}$ is subject to constraints $\gamma_i \in \{0, 1\}$, $\gamma_{2i} + \gamma_{2i-1} \in \{0, 1\}$ and $\sum_{i=1}^m \gamma_i = k$. The number of challenges $|\mathcal{C}|$ equals $\binom{m/2}{k} 2^k$.

The Arbiter PUF of Lee et al. [8] is represented by Fig. 1. A rising edge propagates through two reconfigurable paths with identically designed delays. Because of manufacturing variations however, there is a delay difference Δt between both paths. An arbiter decides which path 'wins' the race ($\Delta t \lessgtr 0$) and generates a response bit r.

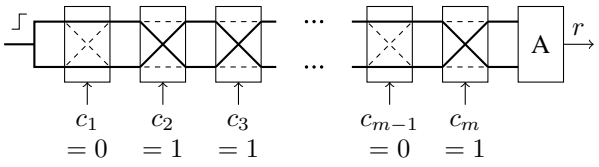

Fig. 1. An Arbiter PUF with m stages.

The two paths are constructed from a series of m switching elements. Challenge bits determine for each stage whether path segments are either crossed or uncrossed. As shown in Fig. 2, each stage has a unique contribution to time difference Δt, depending on its challenge bit. Equation (3) writes Δt as a dot product [2]. The number of challenges $|\mathcal{C}|$ equals 2^m.

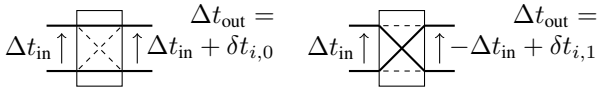

Fig. 2. The delay behavior of a single stage of an Arbiter PUF.

$$\Delta t = \boldsymbol{v}^T \boldsymbol{\gamma}, \quad \text{with } \boldsymbol{v} = \boldsymbol{B}\,\boldsymbol{t}, \boldsymbol{B} = \frac{1}{2}\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & -1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & -1 & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 1 & 1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 & 1 \end{pmatrix},$$
$$\boldsymbol{t} = \begin{pmatrix} \delta t_{1,0} & \delta t_{1,1} & \delta t_{2,0} & \delta t_{2,1} & \dots & \delta t_{m,0} & \delta t_{m,1} \end{pmatrix}^T,$$
$$\text{and } \boldsymbol{\gamma} = \begin{pmatrix} (1-2c_1)(1-2c_2)\dots(1-2c_m) \\ (1-2c_2)\dots(1-2c_m) \\ \vdots \\ (1-2c_m) \\ 1 \end{pmatrix}. \tag{3}$$

To improve the robustness against machine learning, Lee et al. [8] also proposed the feed-forward variant in Fig. 3. Each out of $k \geq 1$ additional arbiter elements thresholds the accumulated time difference after a stage $i \in [1, m-1]$ and drives the challenge bit of a stage $j \in [i+1, m]$. Unfortunately, noise effects are amplified as such. Note that $|\mathcal{C}| = 2^{m-k}$.

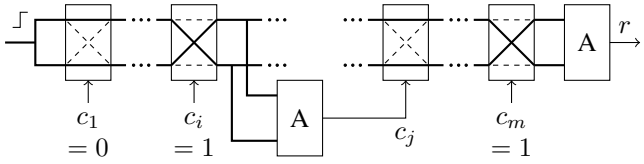

Fig. 3. A Feed-Forward Arbiter PUF with m stages and $k = 1$ loop.

III. First Method: Black-Box Bounds

A first proven method to evaluate upper bounds on the min-entropy of a strong PUF adopts a black-box approach. We rely on machine learning results exclusively, regardless of the internals of the PUF circuit. The method can therefore be applied to simulated PUFs and/or FPGA/ASIC implementations.

A. Theory

The accuracy of a predictive model that relies on g training CRPs is formalized in (4), with $\boldsymbol{n}$ the source of randomness of the possibly randomized training algorithm. Monte Carlo experiments allow for an approximate evaluation of the accuracy, i.e., each expectation deteriorates to an average of a relatively small number of random samples. Normally, $\mathsf{Accuracy}(g) \in [1/2, 1]$ increases monotonically with g. Commonly used techniques include *artificial neural networks*, *logistic regression*, and *evolution strategies*. The learning performance is always suboptimal in practice, i.e., $\mathsf{Accuracy}(g)$ is not the theoretical maximum with respect to a given g, implicating that an upper bound on the min-entropy is the best achievable.

$$\mathsf{Accuracy}(g) = \mathbb{E}_{\boldsymbol{c}_1 \leftarrow C_1}\big[\mathbb{E}_{\boldsymbol{c}_2 \leftarrow C_2}\big[\dots \mathbb{E}_{\boldsymbol{c}_{g+1} \leftarrow C_{g+1}}\big[\mathbb{E}_{\boldsymbol{n} \leftarrow N}\big[\mathbb{E}_{\boldsymbol{v} \leftarrow V}\big[\mathbb{P}(r_{g+1} = \mathsf{Predict}(\boldsymbol{c}_{g+1}; \boldsymbol{c}_1, r_1, \boldsymbol{c}_2, r_2, \dots, \boldsymbol{c}_g, r_g, \boldsymbol{n})\big]\big]\big]\dots\big]\big], \quad \text{with } \forall i \in [1, g+1], r_i \leftarrow \mathsf{PUF}(\boldsymbol{c}_i, \boldsymbol{v}). \tag{4}$$

Consider the device-unique secret $\boldsymbol{x} = r_1 \| r_2 \| \dots \| r_n$ that consists of concatenated response bits. The min-entropy of X obviously depends on the given list of hardcoded challenges, i.e., $\boldsymbol{c}_1$, $\boldsymbol{c}_2$, …, $\boldsymbol{c}_n$. It is our goal however to assess the quality of the PUF in a universal manner, regardless of the implementation details of a particular challenge generator. We therefore resort to the conditional min-entropy in (5), averaging the probability of success of an attacker's best guess over a set of challenge generators. It is fair to assume that challenges are i.i.d. uniform random variables. Challenge generators are typically designed to produce pseudorandom bitstreams whose properties approximate the properties of truly random sequences.

$$\widetilde{\mathbb{H}}_\infty(X|(C_1, C_2, \dots, C_n)) = -\log_2\Big(\mathbb{E}_{\boldsymbol{c}_1 \leftarrow C_1}\big[\mathbb{E}_{\boldsymbol{c}_2 \leftarrow C_2}\big[\dots \mathbb{E}_{\boldsymbol{c}_n \leftarrow C_n}\big[\max_{x \in \mathcal{X}} \mathbb{P}((X = \boldsymbol{x})| ((C_1 = \boldsymbol{c}_1) \cap (C_2 = \boldsymbol{c}_2) \cap \dots \cap (C_n = \boldsymbol{c}_n)))\big]\dots\big]\Big). \tag{5}$$

Consider a possibly randomized procedure $\widehat{\boldsymbol{x}} \leftarrow \mathsf{Guess}(\boldsymbol{c}_1, \boldsymbol{c}_2, \dots, \boldsymbol{c}_n, \boldsymbol{n})$, with $\boldsymbol{n}$ a uniform source of randomness of arbitrary length, used by the attacker to make an educated guess of the PUF response $\boldsymbol{x}$. The probability of success thereof is upper bounded by the optimal guessing procedure, deterministically returning the most likely value of X with respect to the given set of challenges. Performance evaluation of Guess hence results in an upper bound on the min-entropy that is produced by the PUF, as shown in (6).

$$\widetilde{\mathbb{H}}_\infty(X|(C_1, C_2, \dots, C_n)) \leq -\log_2\Big(\mathbb{E}_{\boldsymbol{c}_1 \leftarrow C_1}\big[\mathbb{E}_{\boldsymbol{c}_2 \leftarrow C_2}\big[\dots \mathbb{E}_{\boldsymbol{c}_n \leftarrow C_n}\big[\mathbb{E}_{\boldsymbol{n} \leftarrow N}\big[\mathbb{P}((X = \mathsf{Guess}(\boldsymbol{c}_1, \boldsymbol{c}_2, \dots, \boldsymbol{c}_n, \boldsymbol{n}))| ((C_1 = \boldsymbol{c}_1) \cap (C_2 = \boldsymbol{c}_2) \cap \dots \cap (C_n = \boldsymbol{c}_n)))\big]\big]\dots\big]\big]\Big). \tag{6}$$

Consider the following instantiation of Guess, making use of a machine learning algorithm. The first $g \in [1, n]$ bits of PUF response $\boldsymbol{x}$ are guessed at random, which involves sampling a random variable that is uniform on $\{0,1\}^{g \times 1}$ whenever Predict is evaluated. The success probability thereof is $(1/2)^g$, regardless of the distribution of X. The resulting set of g hopefully correct CRPs comprehends the input of a possibly randomized training algorithm that outputs a model of the PUF. Evaluation of the model provides the last $(n-g)$

bits of prospective response $\widehat{\boldsymbol{x}}$. The upper bound on the min-entropy produced by the PUF reduces to (7).

$$\widetilde{\mathbb{H}}_\infty(X|(C_1,C_2,\ldots,C_n)) \leq -\log_2\Bigg(\mathbb{E}_{\boldsymbol{c}_1\leftarrow C_1}\Bigg[\mathbb{E}_{\boldsymbol{c}_2\leftarrow C_2}\Bigg[\ldots\mathbb{E}_{\boldsymbol{c}_n\leftarrow C_n}\Bigg[\mathbb{E}_{\boldsymbol{n}\leftarrow N}\Bigg[\mathbb{E}_{\boldsymbol{v}\leftarrow V}\Bigg[\Big(\frac{1}{2}\Big)^g \prod_{i=g+1}^{n}\mathbb{P}(X_i=\mathsf{Predict}(\boldsymbol{c}_i;\boldsymbol{c}_1,x_1,\boldsymbol{c}_2,x_2,\ldots,\boldsymbol{c}_g,x_g,\boldsymbol{n})\Bigg]\Bigg]\Bigg]\ldots\Bigg]\Bigg]\Bigg), \quad \text{with } \forall i\in[1,n], x_i\leftarrow\mathsf{PUF}(\boldsymbol{c}_i,\boldsymbol{v}). \tag{7}$$

Given that challenges are independent, the modeling accuracy as defined in (4) reappears. We hence obtain the bound in (8). The value of g that results in the tightest bound is of primary interest. We emphasize that the min-entropy of a sizable multi-bit response X is being bounded here, as this has immediate application to a subsequent fuzzy extractor [3]. A less complicated but also less usable setting bounds the min-entropy of a single response bit only, i.e., $\widetilde{\mathbb{H}}_\infty(R_{g+1}|C_1,R_1,\ldots,C_g,R_g,C_{g+1}) \leq -\log_2(\mathsf{Accuracy}(g))$ [4].

$$\widetilde{\mathbb{H}}_\infty(X|(C_1,C_2,\ldots,C_n)) \leq g-(n-g)\log_2(\mathsf{Accuracy}(g)). \tag{8}$$

B. Numerical Results

Fig. 4 plots the modeling accuracy of RO Sum, S-ArbRO-2, S-ArbRO-4 and Arbiter PUFs with $m = 64$ stages as a function of the number of training CRPs g. To enable comparison with the white-box approach later-on, we use simulated PUFs where frequencies f and delay differences δt are assumed to be independent and identically distributed (i.i.d.) normal random variables. Modeling is greatly facilitated by using $(\boldsymbol{\gamma}_i, r_i)$ as a learning input rather than $(\boldsymbol{c}_i, r_i)$. We use linear regression of an indicator matrix [10] as a classification method. The four learning curves quasi overlap, which is not surprising given that each PUF relies on the thresholding of a dot product.

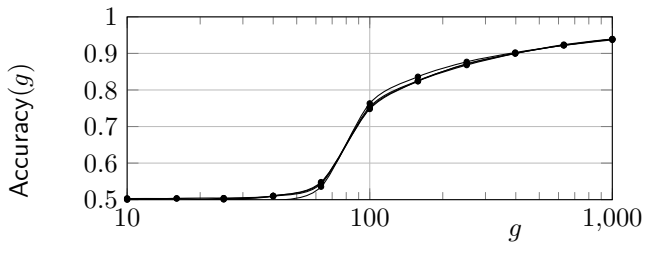

Fig. 4. Machine learning of simulated RO Sum, S-ArbRO-2, S-ArbRO-4 and Arbiter PUFs with $m = 64$. The S-ArbRO-2 and S-ArbRO-4 PUFs are instantiated with $k = 32$ and $k = 16$ respectively, as this maximizes their number of challenges $|\mathcal{C}|$. The modeling accuracy is estimated via Monte Carlo evaluations of size 10^5.

S-ArbRO-2 and S-ArbRO-4 PUFs were claimed to be highly resistant against modeling attacks, especially in comparison to an Arbiter PUF. Our results clearly refute this claim. The authors wrongly assumed that it is necessary to learn a separate model for all possible selections of k contributing stages. Another newly revealed issue, that we do not elaborate further here, is that S-ArbRO-4 PUFs are prone to bias, i.e., randomly generated instances are considerably in favor of producing either 0s or 1s.

Fig. 5 plots upper bounds on the min-entropy of our simulated arbiter PUFs. This result clearly undermines the security of the 128-bit key generator of Van Herrewege et al. [9], relying on an arbiter PUF with $k = 64$ stages equally well. Their 1785-bit response was assumed to be uniform, while our bound implicates that the min-entropy does not exceed 559 bits. Note that we consider simulated PUFs, which differs from the 65nm ASIC manufactured in the research project UNIQUE. Nevertheless, the conclusion hardly changes when plugging in the learning curve of the latter [4].

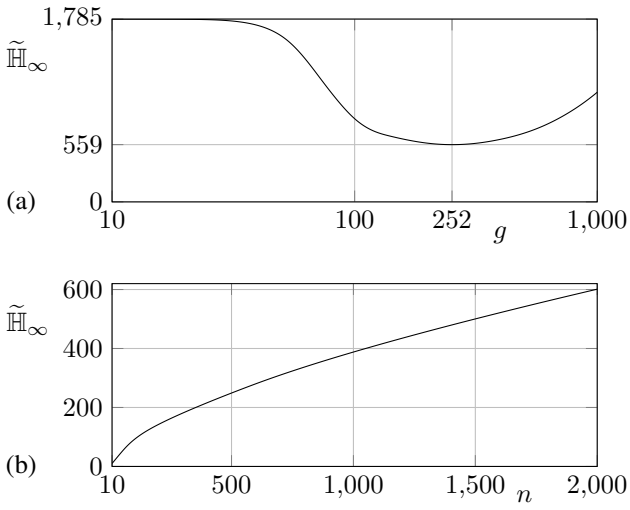

Fig. 5. An upper bound on the min-entropy $\widetilde{\mathbb{H}}_\infty(X|(C_1,C_2,\ldots,C_n))$ of a simulated Arbiter PUF with $m = 64$ stages. (a) For $n = 1785$ challenges, the best bound is approximately 559 bits, hereby using $g = 252$ CRPs as training set. (b) For each response length n, the optimal number of training CRPs g can be selected.

IV. Second Method: White-Box Bounds

The white-box approach requires extensive analysis of the PUF internals, but can result in tighter bounds and improved insights. We derive upper bounds on the min-entropy of the gigantesque $|\mathcal{C}|$-bit concatenated response $\boldsymbol{x}$, obtained by evaluating all possible challenges $\boldsymbol{c}$. The bound is hence valid for all possible challenge generators, typically producing a small subset of $\mathcal{C}$ only. The min-entropy of a random variable, in this case X, can only decrease or remain equal when part of its bits are discarded.

We limit ourselves to ideal PUFs, although it might be feasible to incorporate certain non-idealities that occur in hardware implementations. Frequencies f and delay differences δt are assumed to be i.i.d. normal random variables. Furthermore we assume that frequencies can be measured with infinite

precision, which differs from the digital counters used in hardware. Finally, we consider PUFs to be noiseless, which is not necessarily a far-fetched assumption in practice. Majority voting is a frequently used technique to suppress noise during device enrollment, hereby facilitating key reproduction.

A. RO Sum PUFs

We carry-out the following Monte Carlo experiment on small-scale RO sum PUFs. For a given number of stages $m \in [1,5]$, we evaluate the 2^m-bit concatenated response $\boldsymbol{x}$ of 10^6 randomly generated PUF instances. Test results indicate that the $2m$ outcomes where exactly one v_i dominates are the most likely to occur. For instance, $v_1 > |v_2|+|v_3|+\ldots+|v_m|$, degenerating the thresholding procedure to $\gamma_1\, v_1 \lessgtr 0$. The min-entropy of X is hence upper-bounded as shown in (9), regardless of whether m is small or large. For ease of notation, the pairwise frequency differences are assumed to be standard normal random variables, i.e., $V_i \sim N(0,1)$, with $i \in [1,m]$. The standard deviation can be chosen arbitrarily due to thresholding with 0. We do not formally exclude that another outcome is more likely to occur and therefore claim to have derived a bound rather than an exact result.

$$\mathbb{H}_\infty(X) \leq -\log_2\left(\mathbb{P}\left(V_1 > \sum_{i=2}^{m} |V_i|\right)\right) = -\log_2\left(\int_0^\infty \mathsf{f}_{\mathsf{norm}}(v_1) \int_0^{v_1} 2\mathsf{f}_{\mathsf{norm}}(v_2) \int_0^{v_1-v_2} 2\mathsf{f}_{\mathsf{norm}}(v_3) \ldots \int_0^{v_1-v_2-\ldots-v_{m-1}} 2\mathsf{f}_{\mathsf{norm}}(v_m)\,\mathrm{d}v_m \ldots \mathrm{d}v_3\,\mathrm{d}v_2\,\mathrm{d}v_1\right). \quad (9)$$

Unfortunately, for large m, the nested integrals in (9) cannot be evaluated in a convenient analytical manner. The distribution of the sum of zero-truncated normally distributed random variables is non-trivial. We therefore integrate over a subregion only, which remains consistent with an upper bound on $\mathbb{H}_\infty(X)$. As illustrated in Fig. 6 for $m = 3$, the complete integration domain for v_2 to v_m is bounded by a simplex with vertices $\begin{pmatrix} v_1 & 0 & \ldots & 0 \end{pmatrix}$ to $\begin{pmatrix} 0 & \ldots & 0 & v_1 \end{pmatrix}$. The most straightforward reduction comprehends integrating in a hypercube with vertex $\begin{pmatrix} v_1/(m-1) & v_1/(m-1) & \ldots & v_1/(m-1) \end{pmatrix}$. The resulting expression in (10) can be evaluated easily.

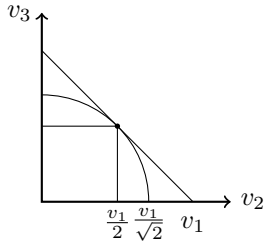

Fig. 6. Reducing the integration domain of (9), illustrated for $m = 3$. Both an inscribed hypercube and an inscribed hypersphere are represented.

$$\mathbb{P}\left(V_1 > \sum_{i=2}^{m} |V_i|\right) \geq \int_0^\infty \mathsf{f}_{\mathsf{norm}}(v_1) \left(\mathsf{F}_{\mathsf{norm}}\left(\frac{v_1}{m-1}\right) - \mathsf{F}_{\mathsf{norm}}\left(-\frac{v_1}{m-1}\right)\right)^{m-1} \mathrm{d}v_1. \quad (10)$$

A tighter upper bound on $\mathbb{H}_\infty(X)$ is obtained by integrating over the inscribed hypersphere instead. A transformation from Cartesian coordinates $\begin{pmatrix} v_2 & v_3 & \ldots & v_m \end{pmatrix}$ to hyperspherical coordinates $\begin{pmatrix} r & \phi_1 & \ldots & \phi_{m-2} \end{pmatrix}$ is defined in (11).

$$\begin{aligned} v_2 &= r\cos(\phi_1), \\ v_3 &= r\sin(\phi_1)\cos(\phi_2), \\ &\vdots \\ v_{m-1} &= r\sin(\phi_1)\ldots\sin(\phi_{m-3})\cos(\phi_{m-2}), \\ v_m &= r\sin(\phi_1)\ldots\sin(\phi_{m-3})\sin(\phi_{m-2}), \\ &\text{with } \phi_1,\ldots,\phi_{m-3} \in [0,\pi) \text{ and } \phi_{m-2} \in [0,2\pi). \end{aligned} \quad (11)$$

As can be derived from the Jacobian determinant, the volume element for integration is $\mathrm{d}v_2\,\mathrm{d}v_3\ldots\mathrm{d}v_m = r^{m-2}\sin^{m-3}(\phi_1)\sin^{m-4}(\phi_2)\ldots\sin(\phi_{m-3})\mathrm{d}r\,\mathrm{d}\phi_1\,\mathrm{d}\phi_2\ldots\mathrm{d}\phi_{m-2}$. We hence obtain (12), valid for $m \geq 3$.

$$\mathbb{P}\left(V_1 > \sum_{i=2}^{m} |V_i|\right) \geq \int_0^\pi \sin^{m-3}(\phi_1)\,\mathrm{d}\phi_1 \int_0^\pi \sin^{m-4}(\phi_2)\,\mathrm{d}\phi_2 \ldots \int_0^\pi \sin(\phi_{m-3})\,\mathrm{d}\phi_{m-3} \int_0^{2\pi} \mathrm{d}\phi_{m-2} \int_0^\infty \mathsf{f}_{\mathsf{norm}}(v_1) \left(\int_0^{\frac{v_1}{\sqrt{m-1}}} \frac{r^{m-2}}{(\sqrt{2\pi})^{m-1}} \exp\left(-\frac{r^2}{2}\right)\mathrm{d}r\right)\mathrm{d}v_1. \quad (12)$$

The latter product of integrals can be fully elaborated. First, observe the recurrence relation in (13).

$$\int_0^\pi \sin^i(\phi)\,\mathrm{d}\phi = \begin{cases} 2 & \text{if } i = 1, \\ \dfrac{\pi}{2} & \text{if } i = 2, \\ \dfrac{i-1}{i}\displaystyle\int_0^\pi \sin^{i-2}(\phi)\,\mathrm{d}\phi & \text{if } i > 2. \end{cases} \quad (13)$$

For m odd, (14) hence holds.

$$\prod_{i=1}^{m-3}\left(\int_0^\pi \sin^i(\phi)\,\mathrm{d}\phi\right) = \frac{\pi^{(m-3)/2}}{(\frac{m-3}{2})!}. \quad (14)$$

Combined with repeated partial integration, we obtain the bound in (15), valid for m odd.

$$\mathbb{H}_\infty(X) \leq -\log_2\left(\frac{1}{2}\left(1 - \sqrt{\frac{m-1}{m}}^{(m-3)/2} \sum_{i=0}^{(m-3)/2} \frac{(2i)!}{(i!)^2(4m)^i}\right)\right). \quad (15)$$

Fig. 7 plots upper bounds on min-entropy using the inscribed hypercube and the inscribed hypersphere respectively. As an additional validation of the latter case, we compute (12) and (15) with MATLAB and Maple respectively, i.e., the same numerical results are produced by two different tools.

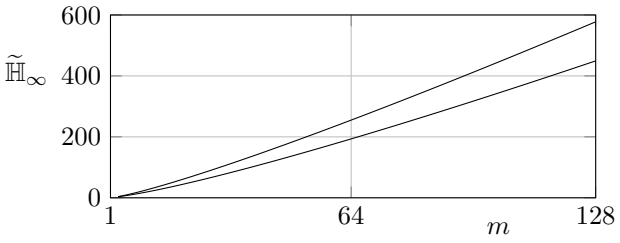

Fig. 7. Upper bounds on the min-entropy of an ideal RO sum PUF as function of the number of stages m. Top and bottom curves correspond to integration via an inscribed hypercube and hypersphere respectively.

B. Arbiter PUF

The upper bound on the min-entropy of an RO Sum PUF with $m+1$ stages applies to an Arbiter PUF with m stages equally well. If $\boldsymbol{v} = \boldsymbol{B}\,\boldsymbol{t}$ is a linear transformation of $T \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then $V \sim N(\boldsymbol{B}\,\boldsymbol{\mu}, \boldsymbol{B}\,\boldsymbol{\Sigma}\,\boldsymbol{B}^T)$. With $\boldsymbol{B}$ defined in (3) and $T \sim N(\boldsymbol{0}, \boldsymbol{I}_{2m})$, it hence holds that $V \sim N(\boldsymbol{0}, \mathsf{diag}(1/2, 1, \ldots, 1, 1/2))$. As all $m+1$ variables are independent, we evaluate a lower-bound on the probability $\mathbb{P}(V_2 > |V_1| + |V_3| + |V_{m+1}|)$ similar to (12). Compared to black-box approach in Fig. 5, the upper bound on the min-entropy is considerable improved. The bound on the 1785-bit response of an arbiter PUF with $m = 64$ stages is reduced from approximately 559 bits to approximately 197 bits.

C. Feed-Forward Arbiter PUF

The upper bound on the min-entropy of an Arbiter PUF with m stages applies to its feed-forward variant with an equal number of stages m stages as well, regardless of the number of loops k and the corresponding tap positions. The main insight is that the feed-forward variant covers only a subset of 2^{m-k} out of 2^m transformed challenges $\boldsymbol{\gamma}$, i.e., part of the gigantesque concatenated response $\boldsymbol{x}$ can be discarded. Despite being more resistant to machine learning [2], the min-entropy hence does not increase, which highlights the superiority of the white-box approach over the black-box approach.

D. S-ArbRO-4 PUF

There is a trivial upper bound on the min-entropy of an S-ArbRO-4 PUF, i.e., $\mathbb{H}_\infty(X) \leq m$. This corresponds to the probability that the signs of all m frequency differences v_i are either all negative or all positive, making concatenated response $\boldsymbol{x}$ either all-zeros or all-ones. Despite outputting up to $\binom{m/2}{k}2^k$ response bits, the produced min-entropy is hence not larger than for the most basic RO PUF design, outputting a short m-bit uniformly distributed response where each v_i is thresholded individually. As a side note, the latter design is a *weak PUF* rather than a strong PUF.

V. Conclusion

The main novelty of this work is the derivation of tight upper bounds on the min-entropy of several strong PUFs. Although not directly usable for the secure instantiation of future key generators, it can show that instantiations from the past are certainly insecure. A side contribution of this work is that S-ArbRO PUFs are shown to be an easy target for machine learning, despite the proposing authors' claim.

VI. Future Work

A suggestion for future work is the security evaluation of the 256-bit key generator of Francq and Parlier [11], using techniques similar to what has been developed in this manuscript. The 440-bit response of a Loop PUF with 64 stages is assumed to provide 418 bits of entropy.

Acknowledgment

The authors greatly appreciate the support received. The European Union's Horizon 2020 research and innovation programme under grant number 644052 (HECTOR). The Research Council of KU Leuven, C16/15/058, the Flemish Government through G.0130.13N and FWO G.0876.14N, and the Hercules Foundation AKUL/11/19. The national major development program for fundamental research of China (973 Plan) under grant number 2013CB338004. Jeroen Delvaux is funded by IWT-Flanders grant number SBO 121552. The authors thank Steven Delvaux for pointing out that (9) can be bounded via an inscribed hypersphere.

References

[1] 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, pp. 26:1–26:42, 2015.

[2] U. Rührmair, J. Sölter, F. Sehnke, X. Xu, A. Mahmoud, V. Stoyanova, G. Dror, J. Schmidhuber, W. Burleson, and S. Devadas, "PUF Modeling Attacks on Simulated and Silicon Data," *IEEE Trans. Information Forensics and Security*, vol. 8, no. 11, pp. 1876–1891, 2013.

[3] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. D. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM J. Comput.*, vol. 38, no. 1, pp. 97–139, 2008.

[4] R. Maes, "Physically Unclonable Functions: Constructions, Properties and Applications," Ph.D. dissertation, KU Leuven, 2012, Ingrid Verbauwhede (promotor).

[5] S. Katzenbeisser, Ü. Kocabaş, V. Rožić, A. Sadeghi, I. Verbauwhede, and C. Wachsmann, "PUFs: Myth, Fact or Busted? A Security Evaluation of Physically Unclonable Functions (PUFs) Cast in Silicon," in *International Workshop on Cryptographic Hardware and Embedded Systems, CHES 2012*, September 2012, pp. 283–301.

[6] M. Yu and S. Devadas, "Recombination of physical unclonable functions," in *Government Microcircuit Applications and Critical Technology Conference, GOMACTech 2010*, March 2010, pp. 1–4.

[7] D. Ganta and L. Nazhandali, "Easy-to-Build Arbiter Physical Unclonable Function with Enhanced Challenge/Response Set," in *International Symposium on Quality Electronic Design, ISQED 2013*, March 2013, pp. 733–738.

[8] J. W. Lee, D. Lim, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas, "A Technique to Build a Secret Key in Integrated Circuits for Identification and Authentication Applications," in *Symposium on VLSI Circuits, Digest of Technical Papers*, June 2004, pp. 176–179.

[9] A. Van Herrewege, S. Katzenbeisser, R. Maes, R. Peeters, A. Sadeghi, I. Verbauwhede, and C. Wachsmann, "Reverse Fuzzy Extractors: Enabling Lightweight Mutual Authentication for PUF-Enabled RFIDs," in *International Conference on Financial Cryptography and Data Security, FC 2012*, February 2012, pp. 374–389.

[10] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer, 2009.

[11] J. Francq and G. Parlier, *Secure Key Generator Using a Loop-PUF*. Cham: Springer International Publishing, 2015, pp. 143–173.