Bias-based modeling and entropy analysis of PUFs

Robbert van den Berg
Eindhoven University of Technology
Eindhoven, The Netherlands

Boris Škorić
Eindhoven University of Technology
Eindhoven, The Netherlands

Vincent van der Leest
Intrinsic-ID
Eindhoven, The Netherlands

ABSTRACT

Physical Unclonable Functions (PUFs) are increasingly becoming a well-known security primitive for secure key storage and anti-counterfeiting. For both applications it is imperative that PUFs provide enough entropy. The aim of this paper is to propose a new model for binary-output PUFs such as SRAM, DFF, Latch and Buskeeper PUFs, and a method to accurately estimate their entropy. In our model the measurable property of a PUF is its set of *cell biases*. We determine an upper bound on the 'extractable entropy', i.e. the number of key bits that can be robustly extracted, by calculating the mutual information between the bias measurements done at enrollment and reconstruction.

In previously known methods only uniqueness was studied using information-theoretic measures, while robustness was typically expressed in terms of error probabilities or distances. It is not always straightforward to use a combination of these two metrics in order to make an informed decision about the performance of different PUF types. Our new approach has the advantage that it simultaneously captures both of properties that are vital for key storage: uniqueness and robustness. Therefore it will be possible to fairly compare performance of PUF implementations using our new method.

Statistical validation of the new methodology shows that it clearly captures both of these properties of PUFs. In other words: if one of these aspects (either uniqueness or robustness) is less than optimal, the extractable entropy decreases. Analysis on a large database of PUF measurement data shows very high entropy for SRAM PUFs, but rather poor results for all other memory-based PUFs in this database.

Categories and Subject Descriptors

B.8.2 [**Hardware**]: Performance and reliability—*Performance Analysis and Design Aids*

Keywords

PUF, SRAM, entropy

TrustED'13, November 4, 2013, Berlin, Germany.

http://dx.doi.org/10.1145/2517300.2517301.

1. INTRODUCTION

Due to deep-submicron manufacturing process variations every transistor in an integrated circuit (IC) has slightly different physical properties that lead to measurable differences. Examples of such physical properties are threshold voltages and gain factors of the IC's transistors. The submicron variations are uncontrollable during manufacturing, which ensures that these physical properties cannot be copied or cloned. Therefore these properties can be used to derive a unique fingerprint of an electronic circuit, similar to human biometrics. It is very hard, expensive and economically not viable to create a device with a specifically chosen fingerprint.

The functions used to derive unique fingerprints for ICs are known as Physical Unclonable Functions (PUFs). Implementing a PUF requires an electronic circuit that measures the responses of the hardware to certain given inputs or challenges, which depend on the unique physical properties of the device. In order for a PUF implementation to be practically useful the PUF should be easy to challenge and the response easy to measure, but very hard to reproduce by construction[1]. Common applications for PUFs are to use them as identification or authentication primitives [6, 11], storing secret keys "without actually storing them" [5, 10] (for IP protection or as a "root of trust" in secure environments), and random number generation [6, 26].

1.1 Physical Unclonable Functions

Pappu [16] introduced the concept of PUFs in 2001 under the name Physical One-Way Functions. The proposed technology was based on obtaining a response (scattering pattern) when shining a laser on a bubble-filled transparent epoxy wafer. In 2002 the first physical random function for silicon devices was introduced by Gassend et al. [4]. This function makes use of the manufacturing process variations in ICs, with identical masks, to uniquely characterize each IC. For this purpose the frequencies of ring oscillators were measured. Using this method (now known as a Ring Oscillator PUF), they were able to characterize ICs. In 2004 Lee et al. [11] proposed another PUF that is based on delay measurements, the Arbiter PUF. In 2010 Suzuki et al. [21] introduced the Glitch PUF, which exploits glitch waveforms from delay variation between gates.

Besides intrinsic PUFs based on delay measurements, a second type of PUF in ICs is known: the memory-based

[1]Note that the way a PUF is implemented is vital to the security of this PUF. E.g. in case of a memory-based PUF there should be no interface on which the start-up pattern can be read by an attacker (PUF response is kept secret).

PUF. These PUFs are based on the measurement of start-up values of memory cells. This memory-based PUF type includes SRAM PUFs, which were introduced by Guajardo et al. in 2007 [5]. Furthermore, so-called Butterfly PUFs were described in 2008 by Kumar et al. [10]. In the same year Maes et al. [14] introduced D Flip-Flop PUFs and Su et al. [19] published about Latch PUFs. Recently, Buskeeper PUFs were demonstrated by Simons et al. [18] in 2012.

1.2 PUF properties

In order for the IC to be uniquely identifiable, the PUF must be reliable and unique. In this case reliable means that one is able to reproduce the same behaviour of the function when challenged with the same input over and over again. The characteristics of electronic components depend on the environment they are exposed to (ambient temperature, voltage ramp-up curves, etc.), but also on the ageing process of CMOS. It is of crucial importance that the function has a stable behaviour across a range of environmental conditions during the lifetime of the IC. Typically it is observed that PUFs exhibit a noisy behaviour. Therefore the PUF implementation must include an error correction process to stabilize the PUF responses both over environmental conditions and over time.

The second important parameter for PUFs is entropy. At the time of PUF manufacture, there is an uncontrollable process that leads to the creation of stably measurable challenge-response properties. The uncontrollability of the manufacturing process ensures the physical unclonability of the PUF, provided that there is enough entropy. We require that the entropy of the uncontrollable stable PUF properties[2] is sufficiently high. When this requirement is met, the following properties hold:

- *Uniqueness.* The probability that two PUFs have closely resembling properties is exponentially small.
- *Unpredictability.* The probability of correctly predicting an unknown PUF's set of responses is exponentially small. Furthermore, knowledge of one PUF's responses does not help in the prediction of another PUF's responses and knowledge of part of PUF response does not help predicting the other bits from this particular response.

1.3 Contribution

The focus of this paper is on demonstrating a novel method for quantifying the usable ('extractable') entropy of PUF responses. *Mutual information* provides a fundamental upper bound on the amount of key material that can be reliably extracted from a PUF using a helper data scheme (a.k.a. Fuzzy Extractor) [3, 8, 15, 22].

We calculate the mutual information between the enrollment measurements and later reconstruction measurements. Here the *bias* of a memory cell / flip-flop / latch serves as the measurable PUF property; multiple enrollment measurements (k) and multiple reconstruction measurements (ℓ) are performed on each cell in order to estimate the bias. The mutual information between the k enrollment measurements and the ℓ reconstruction measurements is an upper bound on the usable entropy.

[2] Entropy of controllable part is irrelevant, since this part can be cloned. Entropy of unstable part is also irrelevant here since we cannot exploit it for reproducible key extraction.

In order to validate our approach and to quantify the results of our approach in a real-life setting we used a large data set from the European project UNIQUE. This statistical validation of the new methodology shows that it clearly captures both the uniqueness and robustness of PUFs. In other words: if one of these aspects is less than optimal, the extractable entropy calculated with this method will decrease. The analysis using the UNIQUE PUF measurement data shows very high entropy for SRAM PUFs, but rather poor results for all other memory-based PUFs of this database.

2. RELATED WORK

This paper has been derived from the work in the M.Sc. thesis of Robbert van den Berg [24] in 2012. In his work a new method is proposed for calculating (extractable) entropy for memory-based PUFs. In this section we briefly list known methods.

Extensive entropy analyses of optical PUFs by Tuyls et al. [23] and of coating PUFs by Škorić et al. [27] exist, but these analyses are not applicable to memory-based PUFs.

A simple first indication of uniqueness involves the calculation of Hamming Weights of PUFs. The Hamming weight of a PUF, the number of cells that return non-zero upon start-up, can be used to determine if a PUF is biased [9, 17, 25]. When sampling multiple PUFs, the minimum or maximum Hamming weight can be used to estimate an upper bound on the bias.

The inter-device (or between-class) Hamming distance is a measure of the uniqueness of PUFs; it indicates how easy it is to distinguish or identify different devices [20]. For uniqueness, it is desirable to have a fractional[3] inter-device distance close to 0.5 which means that on average half the cells prefer a different start-up state [2, 9, 10, 17, 25]. It indicates a low correlation between responses of different devices.

A method to derive a conservative lower bound on the entropy is calculating min-entropy based on the enrollment measurements of a set of PUFs. This is a very conservative entropy estimation, but a good one for measuring uncertainty about a cryptographic key [1, 2, 9, 18, 26]. However, it does not take into account how much entropy is lost due to noise.

An optimal compression algorithm can compress a PUF response to a description with length at least equal to the entropy of the PUF data. By reversing this principle, an optimal compression algorithm can be used to provide an estimate for the PUF entropy. In PUF entropy analysis, the Context-Tree Weighting algorithm (CTW) [28] is regularly used to estimate an upper bound on the entropy of PUFs [1, 2, 7, 17].

Furthermore, in [12] a model was developed for Silicon PUFs, but no entropies were computed. We work with a somewhat similar model and use it to estimate entropies.

3. MODELING BINARY-OUTPUT PUFS

Random variables are written with capitals, and their realizations in lower case. Vectors are in boldface. The number of components (memory bits / flip-flops / latches / ...) in the PUF is denoted as n. The components will be referred to as cells. We define the set $[n] = \{1, \ldots, n\}$. At enrollment, the PUF is fully characterized by a vector of biases: $\boldsymbol{b} = (b_i)_{i=1}^{n}$. When an enrollment measurement is done on cell i, the result

[3] A fractional Hamming distance is the Hamming distance between two strings divided by the length of the strings.

is '1' with probability b_i. For every cell, k enrollment measurements are done (with $k \geq 1$). The number of '1' results in cell i is denoted as x_i. We define $\boldsymbol{x} = (x_i)_{i=1}^n$. The random variable X_i is binomial-distributed with parameters k and b_i: $\Pr[X_i = x] = p_{x|b_i} := \binom{k}{x} b_i^x (1-b_i)^{k-x}$. We denote the joint probability as $p_{\boldsymbol{x}|\boldsymbol{b}} = \prod_{i\in[n]} p_{x_i|b_i}$.

In the reconstruction phase the environmental circumstances are in general different than during enrollment, which leads to modified cell biases b'_i. A number ℓ of measurements is done on each cell; the number of '1' results in cell i is denoted as y_i. The variable Y_i is binomial-distributed with parameters ℓ and b'_i. We define $q_{y|b'_i} = \binom{\ell}{y}(b'_i)^y(1-b'_i)^{\ell-y}$ and $q_{\boldsymbol{y}|\boldsymbol{b}'} = \prod_{i\in[n]} q_{y_i|b'_i}$. Note that $\boldsymbol{x}/k$ is an estimate of $\boldsymbol{b}$, and $\boldsymbol{y}/\ell$ is an estimate of $\boldsymbol{b}'$. The estimates become more accurate by increasing k and ℓ, respectively.

Biases $\boldsymbol{b}$ and $\boldsymbol{b}'$ are themselves the result of probabilistic processes: (i) Random variable $\boldsymbol{B}$ has a distribution ρ dictated by the randomness in PUF manufacturing. (ii) After enrollment there are random influences that alter $\boldsymbol{B}$ to $\boldsymbol{B}'$. This is modeled as a set of transition probabilities $\tau(\boldsymbol{b}'|\boldsymbol{b})$.

The amount of common key material that can be *reliably* extracted from the enrollment and reconstruction measurements is upper bounded by the mutual information $\mathrm{I}(\boldsymbol{X};\boldsymbol{Y}) = \mathsf{H}(\boldsymbol{X}) + \mathsf{H}(\boldsymbol{Y}) - \mathsf{H}(\boldsymbol{X},\boldsymbol{Y})$. Note that $\mathrm{I}(\boldsymbol{X};\boldsymbol{Y})$ depends on k and ℓ. We have

$$\Pr[\boldsymbol{X}=\boldsymbol{x}] = \int_0^1 \mathrm{d}^n b\ \rho(\boldsymbol{b})\ p_{\boldsymbol{x}|\boldsymbol{b}} \quad (1)$$

$$\Pr[\boldsymbol{Y}=\boldsymbol{y}] = \int_0^1 \mathrm{d}^n b' [\int_0^1 \mathrm{d}^n b\ \rho(\boldsymbol{b})\tau(\boldsymbol{b}'|\boldsymbol{b})]\ q_{\boldsymbol{y}|\boldsymbol{b}'} \quad (2)$$

$$\Pr[\boldsymbol{X}=\boldsymbol{x},\boldsymbol{Y}=\boldsymbol{y}] = \int_0^1 \mathrm{d}^n b\ \rho(\boldsymbol{b}) p_{\boldsymbol{x}|\boldsymbol{b}} \int_0^1 \mathrm{d}^n b'\ \tau(\boldsymbol{b}'|\boldsymbol{b}) q_{\boldsymbol{y}|\boldsymbol{b}'} \quad (3)$$

(In our notation the an integral is an operator acting on everything to the right.) Our aim is to estimate ρ and τ from our set of measurements on the UNIQUE PUFs, and then use Eqs. (1–3) to compute $\mathrm{I}(\boldsymbol{X};\boldsymbol{Y})$. However, the space in which the biases live is very large due to the large number of cells ($\boldsymbol{b}, \boldsymbol{b}' \in \mathcal{B} = [0,1]^n$), no matter how we discretize the interval $[0,1]$. This makes estimation of probability distributions difficult, since any histogram we construct is based on only N points in the whole space $\mathcal{B}$, where N is the number of PUFs we have at our disposal; the density of points is so low that typically each bin will contain at most one point.

We introduce the following, rather crude, approximation,

$$\rho(\boldsymbol{b}) \approx \prod_{i\in[n]} \rho_i(b_i) \quad ; \quad \tau(\boldsymbol{b}'|\boldsymbol{b}) \approx \prod_{i\in[n]} \tau_0(b'_i|b_i). \quad (4)$$

In words: (i) At manufacture, each cell has its own probability distribution (ρ_i) for the bias, independent of the other cells. (ii) We use *global* transition probabilities $\tau_0(\ \cdot\ |\ \cdot\ )$, independent of the cell index, to model the effect of environmental influences on the biases.

The functions ρ_i and τ_0 are defined on small domains: $[0,1]$ and $[0,1]^2$ respectively. Hence they can be estimated fairly accurately. Note that our approximation for ρ is not capable of modeling correlations between cells. Our approach (4) is motivated by (a) the lack of correlation we observe between cells in most of the PUF types (see Section 4.3), and (b) a feeling that the physics of the transitions $b_i \mapsto b'_i$ should not depend on the cell index i.

Substitution of (4) into (1–3) gives factorized equations,

$$\Pr[\boldsymbol{X}=\boldsymbol{x}] \approx \prod_{i\in[n]} \int_0^1 \mathrm{d}b_i\ \rho_i(b_i) p_{x_i|b_i} \quad (5)$$

$$\Pr[\boldsymbol{Y}=\boldsymbol{y}] \approx \prod_{i\in[n]} \int_0^1 \mathrm{d}b'_i [\int_0^1 \mathrm{d}b_i\ \rho_i(b_i)\tau_0(b'_i|b_i)] q_{y_i|b'_i} \quad (6)$$

$$\Pr[\boldsymbol{X}=\boldsymbol{x},\boldsymbol{Y}=\boldsymbol{y}] \approx \prod_{i\in[n]} \int_0^1 \mathrm{d}b_i \rho_i(b_i) p_{x_i|b_i} \int_0^1 \mathrm{d}b'_i \tau_0(b'_i|b_i) q_{y_i|b'_i} \quad (7)$$

The mutual information then consists of independent parts, $\mathrm{I}(\boldsymbol{X};\boldsymbol{Y}) \approx \sum_{i\in[n]} \mathrm{I}(X_i;Y_i) = \sum_{i\in[n]} \mathsf{H}(X_i) + \mathsf{H}(Y_i) - \mathsf{H}(X_i,Y_i)$.

4. RESULTS

4.1 Data set

To test the results of our proposed method, a large data set of PUF measurements has been used. This data set was created in the EU funded FP7 programme project UNIQUE (contract number 238811). The UNIQUE project yielded 192 ASICs featuring six different PUF types: SRAM, D Flip-Flop (DFF), Latch, Buskeeper, Arbiter and Ring Oscillator.

We analyze the four memory-based PUF types. Each ASIC has four instantiations of the Latch, DFF and SRAM PUF and two instantiation of the Buskeeper PUF. Unfortunately two Latch PUFs per ASIC are unusable due to faults in the addressing logic. Furthermore, during preliminary testing, one DFF instance was found to be very unreliable when compared to the other instances (also noted in [9, 13]). This instance we also excluded from the test data. This leaves a total of $2 \cdot 192 = 384$ Latch and Buskeeper PUFs, $3 \cdot 192 = 576$ DFF and $4 \cdot 192 = 768$ SRAM PUFs for analysis.

All these PUF types provide 8192 bits of output, except the SRAM PUF which has 65536 bits of output. However, during the entropy analysis we used only 8192 out of these 65536, in order to reduce the required memory for processing.

In the UNIQUE project, several different test (such as temperature and voltage variation) were done to determine reliability (e.g. in [9] and [12]). In this paper, we use the data from the temperature variation test for the uniqueness analysis, since it provides PUF responses obtained both at room temperature and at the standard operational temperature limits of electronics. The room temperature measurements are ideal candidates for enrollment, while the measurements at other temperatures provide reconstruction conditions. The data set contains a total of 60 measurements per PUF instantiation at +25°C (used as enrollment measurements) and 40 measurements at -40°C and +85°C respectively (used as reconstruction measurements). The two temperatures used for reconstruction have been chosen because the industrial standard for temperature testing of ICs ranges from -40°C to +85°C. Therefore, these two temperatures are the corner cases for using PUFs in industrial grade devices.

The analysis of the data has been performed on a 32-bit 3GHz dual core PC with 2GB RAM, using Matlab. To process the PUF data with Matlab, data matrices were created with cells as columns and measurements as rows. This was repeated for each PUF, creating a #Measurements × #Cells × #PUFs three-dimensional matrix per PUF instance. The memory size required for these matrices can become rather large as the number of elements of these matrices increases. For example, if all cells of the SRAM PUF would be used, a

$60 \times 65536 \times 192 = 754{,}974{,}720$ element matrix would be required to store enrollment data. However, as some Matlab functionality only works with (64-bit) doubles, these matrices cannot be stored and processed efficiently.

4.2 Applying the proposed model

In order to apply the model as proposed in Section 3 we need to investigate whether individual PUF cells are correlated with each other. Since the proposed method requires PUF cells to be independent, it should be made sure that this is indeed the case for the PUFs from the UNIQUE database. This verification is described in Section 4.3.

In order to make contact with the approaches in the literature, we first separately investigate PUF uniqueness and reliability, before presenting the mutual information results. A measure of device uniqueness is calculated in Section 4.4. For this purpose we use the inter-device distance. As stated before, the extractable entropy derived by the proposed model is based, besides uniqueness, also on the reliability of the PUFs. The robustness of the biases is calculated in Section 4.5.

We compute the mutual information in Section 4.6. This mutual information contains aspects of both the uniqueness and the reliability. The mutual information computed according to our model provides an estimated upper bound on the extractable entropy per cell. Finally we calculate the amount of extractable entropy per mm^2 for each PUF type.

Note that all results in this paper are taken from the M.Sc. thesis of Robbert van den Berg [24]. For more details on the results and for comparisons of our method with results from methods in literature, we refer the reader to this thesis.

4.3 Correlations

Pearson's product-moment coefficient is calculated for every PUF to determine if there is any correlation among cells. Although 0 correlation does not directly imply independence, any correlation found during testing would indicate that there exists linear dependencies between cells. In literature, PUF cells are generally assumed independent (e.g. [2, 18]).

For this test, the first 1024 cells of each PUF are used. Pairwise, the covariance of two cells is divided by the product of their standard deviations as shown in Eq. (8). The result is a value between -1 and 1, where 1 denotes a very strong positive relation and -1 denotes a very strong negative relation, which means that when the bias of cell i increases, the bias of cell j decreases. The closer this value lies to zero the weaker the relationship between the two cells.

$$\mathrm{Corr}(x_i, x_j) = \frac{\mathrm{Cov}(x_i, x_j)}{\sigma_{x_i}\sigma_{x_j}} \tag{8}$$

Furthermore, we calculated the probabilities of getting a correlation as large as observed under the hypothesis that there is no correlation. When this probability is less than 0.01, a correlation is considered significant.

For all PUF instances, the percentage of cells failing the hypotheses of no correlation lies around 0.010 with a maximum of 0.014 for the Latch PUF. This amount of significant correlations is exactly what can be expected by chance. Furthermore, from the significant correlations, the strength of the correlation is approximately 0.2. When the same correlation test is applied to synthetically generated PUF data (known to be independent), similar values are observed. These results indicate that linear dependence among cells is very low or non-existent. We cannot exclude nonlinear dependences.

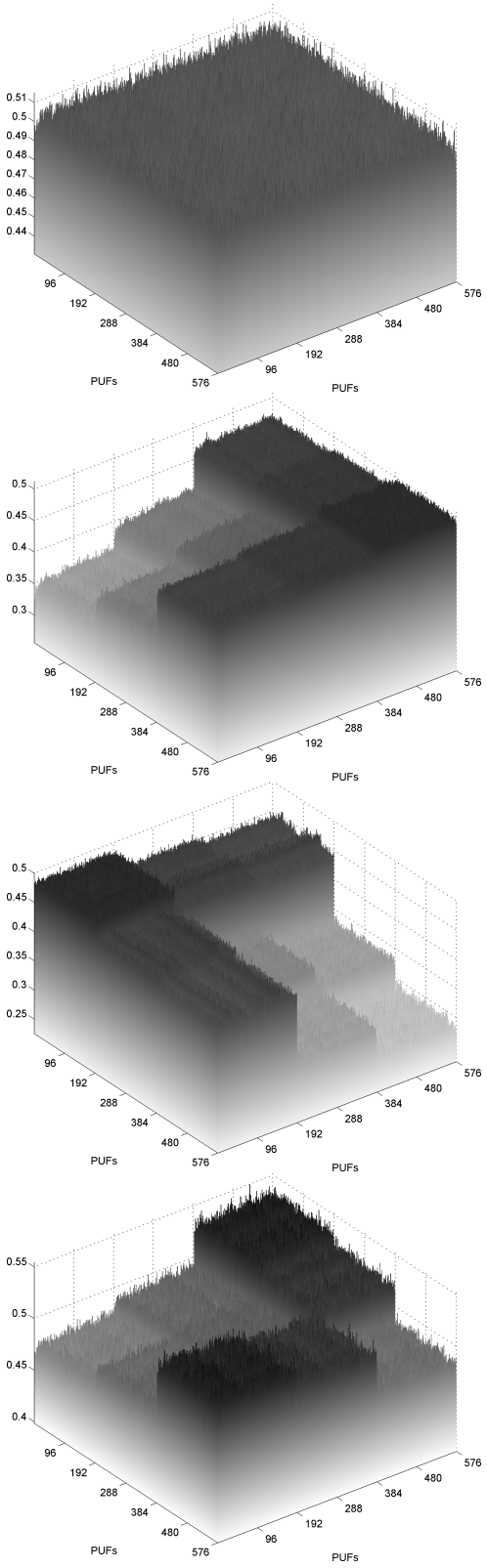

Figure 1: *Inter-device distances $\Delta_{p,p'}$. Device numbers 1–192 refer to the set of devices at $-40°C$, 193–384 denote the same devices at $+25°C$, and 385–576 at $+85°C$.* **From top to bottom:** *SRAM, DFF, Latch and Buskeeper.*

4.4 Inter-device distance

Let $x_i^{(p)}$ be the x-count in cell i of PUF p. We define the inter-device distance $\Delta_{p,p'}$ between PUFs p and p' as the cell-average of the absolute bias difference,

$$\Delta_{p,p'} := \frac{1}{n}\sum_{i=1}^{n}\left|\frac{x_i^{(p)}}{k} - \frac{x_i^{(p')}}{k}\right|. \tag{9}$$

Fig. 1 shows inter-device distances. Here the device numbers 1–192 refer to the set of devices at −40°C, while 193–384 denote the same set of devices at +25°C, and 385–576 at +85°C. For the SRAM PUFs the temperature seems to have no effect on the inter-device distances, which are all close to 50%. This indicates a close to optimal inter-device distance (around 50% is optimal), which is also very stable over different environmental conditions.

For DFF PUFs the distances become smaller with decreasing temperature. This happens because the average Hamming Weight of the DFF PUFs rises with decreasing temperature. At −40°C this value gets close to 100%, which leaves little room for differences between devices.

In Latch PUFs the opposite happens: distances become smaller with increasing temperature. In this case Hamming Weight rises with temperature (close to 100% at +85°C).

The Buskeeper behaves differently. There is a marked difference between +85°C and the other temperatures. This is because the Buskeeper memories are slightly biased towards 0 at −40° and +25°, while there is a significant bias towards 1 at +85°. The result is a lower inter-device distance when comparing devices at an equal temperature and comparing −40° to +25°. Comparing devices at +85° to any other temperature results in a higher inter-device distance, since the Hamming Weight is very different in these cases.

4.5 Robustness of the biases

We denote the vector $\boldsymbol{x}$ associated with the a'th PUF as $\boldsymbol{x}^{(a)}$, and similarly $\boldsymbol{y}^{(a)}$. The robustness of a cell's bias can be characterized using the following distance measure,

$$D_i := \frac{1}{N}\sum_{a=1}^{N}\left|\frac{x_i^{(a)}}{k} - \frac{y_i^{(a)}}{\ell}\right|. \tag{10}$$

Here $i \in [n]$ is the cell index. Values for large k and ℓ are listed in Table I.

Table I. *Bias robustness of UNIQUE PUFs. Listed values are average and maximum D_i values, with k and ℓ very large.*

	Av. distance		*Max. distance*	
Instance	-40°C	+85°C	-40°C	+85°C
SRAM 1	0.054	0.050	0.059	0.056
SRAM 2	0.053	0.050	0.060	0.057
SRAM 3	0.053	0.050	0.059	0.057
SRAM 4	0.053	0.050	0.061	0.058
DFF 1	0.125	0.176	0.158	0.194
DFF 2	0.153	0.174	0.318	0.217
DFF 3	0.122	0.178	0.157	0.196
DFF 4	0.120	0.177	0.166	0.197
Latch 1	0.231	0.103	0.274	0.171
Latch 2	0.233	0.117	0.277	0.182
Buskeeper 1	0.09	0.172	0.099	0.196
Buskeeper 2	0.092	0.171	0.101	0.20

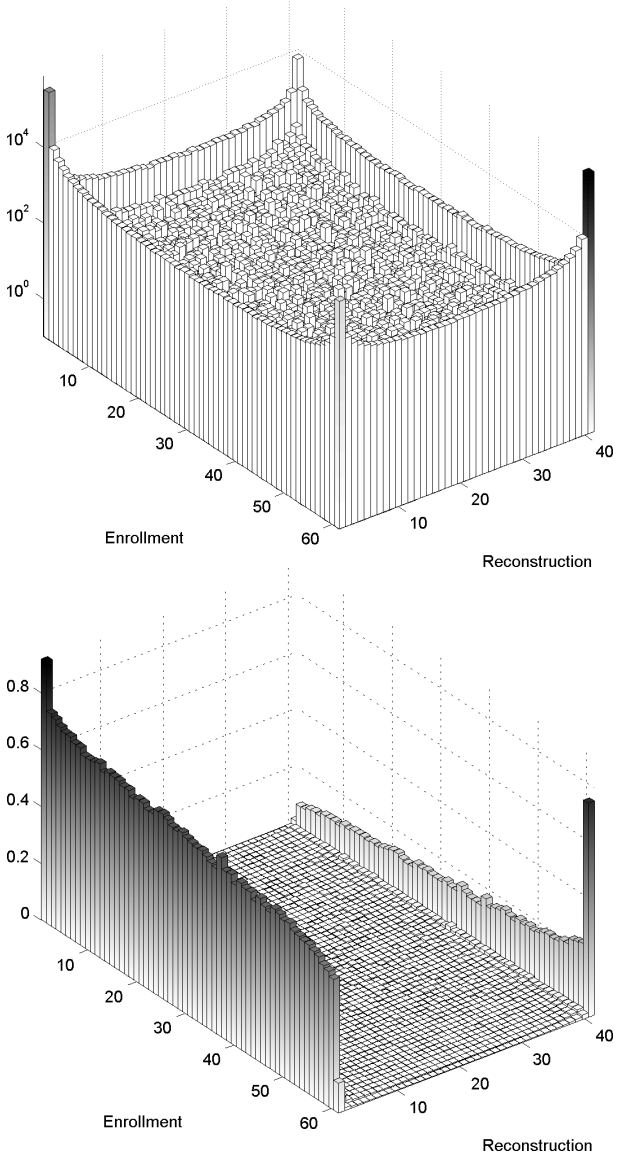

Figure 2: *Bias changes in the DFF PUFs at +85°C.* **Top:** *Histogram of (x_i, y_i) pairs, for $k = 60$, $\ell = 40$, on a logarithmic scale. The vertical axis counts the number of cells in which a combination (x, y) occurs.* **Bottom:** *The transition probabilities $\tau_0(\frac{y}{\ell}|\frac{x}{k})$ derived from the histogram, plotted as a function of x and y.*

In Fig. 2 we show observed bias transition counts and the transition model derived from them (transition probabilities τ_0). The figure shows the result for DFF PUFs; the other PUF types behave similarly. We see that biases far away from 0 and 1 practically never occur (not even when the enrollment bias lies around 0.5). Note that the top figure is logarithmically scaled in order to make the low parts of the histogram visible. Hence, the typical bias changes that occur are jumps to 0 or 1.

Furthermore, as expected, in the bottom figure we see that the probability mass of y given x is concentrated at small y when x is small, and at large y when x is large. In DFF PUFs at +85°C, bias jumps to 0 are more likely than jumps to 1.

4.6 Mutual information

For each of the four PUF types we have estimated the mutual information $I(\boldsymbol{X};\boldsymbol{Y})$ using the independent-cell approximation (4), with empirical ρ_i and τ_0. The results are shown in Fig. 3, as an average per cell, as a function of k and ℓ. Unsurprisingly, (i) the mutual information grows with increasing k and ℓ; and (ii) saturation occurs at large k,ℓ.

The rate of growth is not the same for all PUF types. SRAM PUFs benefit most from increasing k and ℓ. Note that SRAM PUFs can achieve a mutual information of more than one bit per cell. This is entirely natural, since this mutual information is calculated based on the values of the cell biases (and not on the binary start-up values of these cells). These cell biases are continuum variables which in theory can have infinite entropy. Note also that even at $k = 1$ (a single enrollment measurement) it is advantageous to take $\ell > 1$.

Finally, based on the mutual entropy results and known size of the PUF instances on the UNIQUE ASIC (based on [13]) the minimum number of extractable bits per mm^2 of each PUF type can be calculated. The results of the calculation can be found in Table III.

From these results it becomes very clear that the SRAM PUF by far has the highest extractable entropy out of all these PUF types. This is no surprise, since SRAM PUFs were found to be the most reliable and unique PUFs in [9, 13]. Furthermore, the number of PUF cells per mm^2 is also highest for the SRAM PUF. Hence there are multiple reasons why none of the other PUFs even comes close to the performance of the SRAM PUF.

Out of the other (memory-based) PUF types, the Buskeeper PUF can be ranked in second place (fairly good uniqueness, but much less robust over temperature variations). Both the DFF and Latch PUFs (ranked third and fourth respectively) perform much worse, because for these PUFs both the uniqueness and robustness are poor. All of these results are comparable to the conclusions drawn in [9, 13] about the UNIQUE data set.

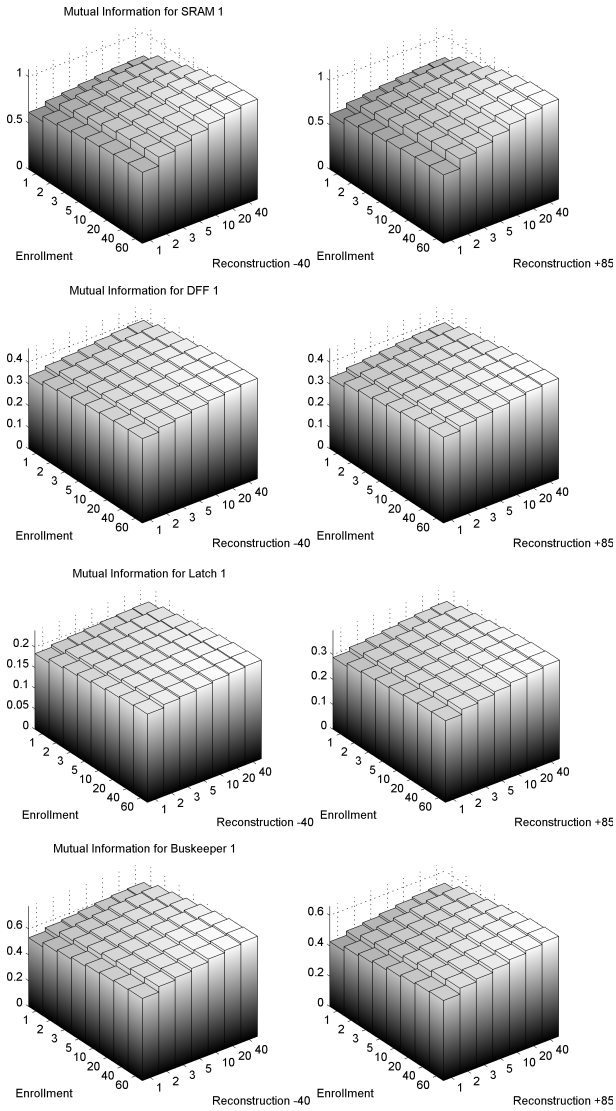

Figure 3: *Mutual information between X_i and Y_i as a function of k and ℓ, for the four PUF types, at reconstruction temperatures $-40°C$ and $+85°C$.* **From top to bottom:** *SRAM, DFF, Latch and Buskeeper.*

Table II. *Mutual information for different k and ℓ. Instance and condition giving the lowest mutual information per PUF type is marked with *.*

Cond.	Instance	Mutual Information (per cell)		
		k=1, ℓ=1	k=60, ℓ=1	k=60, ℓ=40
−40°C	SRAM 1*	0.61	0.77	1.08
	SRAM 2	0.61	0.77	1.08
	SRAM 3	0.61	0.77	1.08
	SRAM 4	0.62	0.77	1.08
	DFF 1	0.33	0.39	0.46
	DFF 3	0.33	0.39	0.45
	DFF 4*	0.33	0.38	0.44
	Latch 1*	0.18	0.21	0.24
	Latch 2	0.20	0.23	0.26
	Busk. 1	0.53	0.63	0.77
	Busk. 2	0.52	0.62	0.75
+85°C	SRAM 1	0.62	0.77	1.12
	SRAM 2	0.62	0.77	1.12
	SRAM 3	0.62	0.77	1.12
	SRAM 4	0.62	0.77	1.12
	DFF 1	0.33	0.40	0.46
	DFF 3	0.32	0.39	0.46
	DFF 4	0.33	0.40	0.46
	Latch 1	0.29	0.33	0.40
	Latch 2	0.28	0.32	0.39
	Busk. 1	0.43	0.53	0.66
	Busk. 2*	0.42	0.52	0.65

Table III. *Extractable bits per mm^2 on the UNIQUE chip, broken down to PUF type and depending on k and ℓ. The lowest numbers were taken from Table II.*

PUF type	Area (mm^2)	Cells/ mm^2	Minimum #bits/mm^2		
			k=1, ℓ=1	k=60, ℓ=1	k=60, ℓ=40
SRAM	0.213	≈ 1.2M	0.75M	0.95M	1.3M
DFF	0.392	≈ 84k	28k	32k	37k
Latch	0.272	≈ 0.12M	22k	25k	29k
Busk.	0.076	≈ 0.22M	91k	0.11M	0.14M

5. CONCLUSIONS AND FUTURE WORK

We have developed a model for memory-based PUFs that treats the *cell biases as the identifying property* of the PUF. The enrollment procedure, consisting of k measurements, gives an estimate $\boldsymbol{X}/k$ of the cell biases $\boldsymbol{b}$ under enrollment conditions; similarly the ℓ reconstruction measurements give an estimate $\boldsymbol{Y}/\ell$ of the biases $\boldsymbol{b}'$ at reconstruction conditions. The mutual information $I(\boldsymbol{X};\boldsymbol{Y})$ is an upper bound on the amount of key material that can be reliably extracted from the PUF. The mutual information depends on the probability distribution $\rho(\boldsymbol{b})$, which models the uncontrollable manufacturing process, and on the transition probabilities $\tau(\boldsymbol{b}'|\boldsymbol{b})$ which model the various sources of noise.

This approach has the advantage that it simultaneously captures two issues of vital importance for key storage: uniqueness and robustness. (Usually only uniqueness is studied using information-theoretic measures; robustness is typically expressed in terms of error probabilities or distances.)

We have applied our model to the UNIQUE date set, assuming that all cells are independent. Furthermore, we have adopted a specific noise model in which the transition probabilities $\tau_0(b'|b)$ do not depend on the cell index. Our analysis shows a very high entropy for the SRAM PUFs in the UNIQUE database (especially when the number of enrollment and reconstruction measurements increases, the entropy per cell becomes more than 1). However, all other PUFs contain significantly less entropy. The Latch PUFs perform poorly, with values between 0.18 and 0.40 bits of entropy per cell.

Based on the results from this paper, we foresee as future work:

- Mutual information estimates including correlations between cells. This requires dealing with an $n \times n$ correlation matrix, which is cumbersome for large n.
- We have not addressed the question of Fuzzy Extractor design. The mutual information $I(\boldsymbol{X};\boldsymbol{Y})$ is an upper bound on the amount of extractable key material, but knowing this number does not tell you *how* to achieve this bound. Efficient Fuzzy Extractors have to be found.

Acknowledgements

This work has been supported by the European Commission through the ICT program under contract INFSO-ICT-284833 (PUFFIN).

References

[1] Frederik Armknecht, Roel Maes, Ahmad-Reza Sadeghi, Franccois-Xavier Standaert, and Christian Wachsmann. 2011. A Formal Foundation for the Security Features of Physical Functions. *IEEE Security and Privacy 2011* 2011, 1 (2011), 16.

[2] Mathias Claes, Vincent van der Leest, and An Braeken. 2012. Comparison of SRAM and FF PUF in 65nm technology. In *Proceedings of the 16th Nordic conference on Information Security Technology for Applications (NordSec'11)*. Springer-Verlag, Berlin, Heidelberg, 47–64. DOI: http://dx.doi.org/10.1007/978-3-642-29615-4_5

[3] Y. Dodis, M. Reyzin, and A. Smith. 2004. Fuzzy Extractors: How to generate strong keys from biometrics and other noisy data. In *Eurocrypt 2004 (LNCS)*, Vol. 3027. 523–540.

[4] Blaise Gassend, Dwaine Clarke, Marten van Dijk, and Srinivas Devadas. 2002. Silicon physical random functions. In *ACM Conference on Computer and Communications Security (CCS'02)*. ACM, New York, NY, USA, 148–160.

[5] Jorge Guajardo, Sandeep S. Kumar, Geert-Jan Schrijen, and Pim Tuyls. 2007. FPGA Intrinsic PUFs and Their Use for IP Protection. In *Workshop on Cryptographic Hardware and Embedded Systems (CHES '07) (LNCS)*, Pascal Paillier and Ingrid Verbauwhede (Eds.), Vol. 4727. Springer-Verlag, Berlin, Heidelberg, 63–80. DOI:http://dx.doi.org/10.1007/978-3-540-74735-2_5

[6] Daniel E. Holcomb, Wayne P. Burleson, and Kevin Fu. 2009. Power-Up SRAM State as an Identifying Fingerprint and Source of True Random Numbers. *IEEE Trans. Computers* 58, 9 (2009), 1198–1210.

[7] Tanya Ignatenko, Geert-Jan Schrijen, Boris Škorić, Pim Tuyls, and Frans M. J. Willems. 2006. Estimating the secrecy rate of Physical Uncloneable Functions with the Context-Tree Weighting method. In *Proc. IEEE International Symposium on Information Theory 2006*. Seattle, USA, 499–503.

[8] A. Juels and M. Wattenberg. 1999. A fuzzy commitment scheme. In *ACM Conference on Computer and Communications Security (CCS) 1999*. 28–36.

[9] Stefan Katzenbeisser, Ünal KocabaÅ§, Vladimir Rožić, Ahmad-Reza Sadeghi, Ingrid Verbauwhede, and Christian Wachsmann. 2012. PUFs: Myth, Fact or Busted? A Security Evaluation of Physically Unclonable Functions (PUFs) Cast in Silicon. In *Cryptographic Hardware and Embedded Systems (CHES) 2012*, Emmanuel Prouff and Patrick Schaumont (Eds.). Lecture Notes in Computer Science, Vol. 7428. Springer Berlin Heidelberg, 283–301. DOI:http://dx.doi.org/10.1007/978-3-642-33027-8_17

[10] S.S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, and P. Tuyls. 2008. The butterfly PUF protecting IP on every FPGA. In *IEEE International Workshop on Hardware-Oriented Security and Trust (HOST'08)*, Mohammad Tehranipoor and Jim Plusquellic (Eds.). IEEE Computer Society, 67–70. DOI:http://dx.doi.org/10.1109/HST.2008.4559053

[11] J.W. Lee, Daihyun Lim, B. Gassend, G.E. Suh, M. van Dijk, and S. Devadas. 2004. A technique to build a secret key in integrated circuits for identification and authentication applications. In *IEEE Symposium on VLSI Circuits 2004*. IEEE, 176–179. DOI:http://dx.doi.org/10.1109/VLSIC.2004.1346548

[12] R. Maes. 2013. An Accurate Probabilistic Reliability Model for Silicon PUFs. In *Workshop on Cryptographic Hardware and Embedded Systems (CHES) 2013*.

[13] R. Maes, V. Rozic, I. Verbauwhede, P. Koeberl, E. van der Sluis, and V. van der Leest. 2012. Experimental evaluation of Physically Unclonable Functions in 65 nm CMOS. In *ESSCIRC (ESSCIRC), 2012 Proceedings of the*. 486 –489. DOI:http://dx.doi.org/10.1109/ESSCIRC.2012.6341361

[14] Roel Maes, Pim Tuyls, and Ingrid Verbauwhede. 2008. Intrinsic PUFs from Flip-flops on Reconfigurable Devices. In *Workshop on Information and System Security (WISSec 2008)*. Eindhoven, NL, 17.

[15] J.-P. Linnartz P. and Tuyls. 2003. New Shielding Functions to Enhance Privacy and Prevent Misuse of Biometric Templates. In *Audio- and Video-Based Biometric Person Authentication*. Springer.

[16] Ravikanth Srinivasa Pappu. 2001. *Physical one-way functions*. Ph.D. Dissertation. Massachusetts Institute of Technology. AAI0803255.

[17] Geert-Jan Schrijen and Vincent van der Leest. 2012. Comparative analysis of SRAM memories used as PUF primitives. In *Design, Automation Test in Europe Conference Exhibition (DATE) 2012*. 1319 –1324.

[18] Peter Simons, Erik van der Sluis, and Vincent van der Leest. 2012. Buskeeper PUFs, a Promising Alternative to D Flip-Flop PUFs. In *IEEE International Workshop on Hardware-Oriented Security and Trust (HOST'12), in print*. IEEE Computer Society.

[19] Ying Su, J. Holleman, and B.P. Otis. 2008. A Digital 1.6 pJ/bit Chip Identification Circuit Using Process Variations. *Solid-State Circuits, IEEE Journal of* 43, 1 (2008), 69–77. DOI:http://dx.doi.org/10.1109/JSSC.2007.910961

[20] G.E. Suh and S Devadas. 2007. Physical Unclonable Functions for Device Authentication and Secret Key Generation. In *Design Automation Conference, 2007. DAC '07. 44th ACM/IEEE*. 9–14.

[21] Daisuke Suzuki and Koichi Shimizu. 2010. The Glitch PUF: A New Delay-PUF Architecture Exploiting Glitch Shapes. In *Cryptographic Hardware and Embedded Systems, CHES 2010*, Stefan Mangard and Francois-Xavier Standaert (Eds.). Lecture Notes in Computer Science, Vol. 6225. Springer Berlin Heidelberg, 366–382. DOI:http://dx.doi.org/10.1007/978-3-642-15031-9_25

[22] P. Tuyls, B. Škorić, and T. Kevenaar. 2007. *Security with Noisy Data: Private Biometrics, Secure Key Storage and Anti-Counterfeiting*. Springer, London.

[23] P. Tuyls, B. Škorić, S. Stallinga, T. Akkermans, and W. Ophey. 2004. An information theoretic model for Physical Uncloneable Functions. In *Information Theory, 2004. ISIT 2004. Proceedings. International Symposium on*. 141–. DOI:http://dx.doi.org/10.1109/ISIT.2004.1365176

[24] R. van den Berg. 2012. Entropy analysis of Physical Unclonable Functions. MSc. thesis, Eindhoven University of Technology. (2012).

[25] Vincent van der Leest, Geert-Jan Schrijen, Helena Handschuh, and Pim Tuyls. 2010. Hardware intrinsic security from D flip-flops. In *Proceedings of the fifth ACM workshop on Scalable trusted computing (STC '10)*. ACM, New York, NY, USA, 53–62. DOI:http://dx.doi.org/10.1145/1867635.1867644

[26] Vincent van der Leest, Erik van der Sluis, Geert-Jan Schrijen, Pim Tuyls, and Helena Handschuh. 2012. Efficient Implementation of True Random Number Generator Based on SRAM PUFs. In *Cryptography and Security: From Theory to Applications*, David Naccache (Ed.). Lecture Notes in Computer Science, Vol. 6805. Springer Berlin Heidelberg, 300–318.

[27] B. Škorić, S. Maubach, T. Kevenaar, and P. Tuyls. 2006. Information-theoretic analysis of capacitive Physical Unclonable Functions. *Journal of Applied Physics* 100, 2 (2006), 024902–024902–11. DOI:http://dx.doi.org/10.1063/1.2209532

[28] F.M.J. Willems, Y.M. Shtarkov, and T.J. Tjalkens. 1995. The context-tree weighting method: basic properties. *Information Theory, IEEE Transactions on* 41, 3 (1995), 653–664. DOI:http://dx.doi.org/10.1109/18.382012