Eighth International Workshop on Optimal Codes and Related Topics
July 10-14, 2017, Sofia, Bulgaria *pp. 98-103*

Applying convolutional codes to key extraction using ring oscillator PUFs

Sven Müelich sven.mueelich@uni-ulm.de
Martin Bossert martin.bossert@uni-ulm.de
Institute of Communications Engineering, University of Ulm, Germany

Abstract. Recently, convolutional codes were applied to error correction in Physical Unclonable Functions (PUFs). In previous work, we used a mathematical model from the literature in order to obtain reliability values for the several bits extracted from an SRAM PUF. In this work we use real world data which are available for a collection of Ring Oscillator PUFs (ROPUFs), in order to verify that the codes suggested based on the mathematical model are also practicable.

1 Motivation

Physical Unclonable Functions (PUFs) can be applied in order to generate bit sequences that can be used for cryptographic applications like secure key generation. Since extracted bit sequences can be reproduced when needed, keys do not have to be stored in non-volatile memory. An important component of the reproduction process is error correction, since the bit sequences are not perfectly reproducible. Recently, we used convolutional codes for error correction in PUFs [6]. Instead of applying a mathematical model, this work uses real world data, obtained by on-chip measurements from Ring Oscillator PUFs (ROPUFs). We use a huge dataset consisting of ring oscillator frequencies, which was developed in [5] and provided for public usage.

The contributions of this work are: First, we verify that the codes suggested in [6] are applicable for practical PUFs. Second, the codes can be seen as a suggestion of how error correction can be added to the PUFs in [5].

2 Physical Unclonable Functions (PUFs)

A PUF is an unclonable physical device from which a unique and reproducible random bit sequence (response) can be extracted. The randomness is intrinsic to the device due to variations within the manufacturing process. Responses can be used as cryptographic keys or as unique identifiers for the devices. However, when reproducing a response the results vary, since PUFs are sensitive to environmental conditions like temperature, supply voltage or aging. Error correction compensates this effect.

In this work we focus on *Ring Oscillator PUFs (ROPUFs)*, introduced in [7], due to available real world data. A ring oscillator (RO) oscillates with a

frequency which depends on delays in inverters and wires of the RO. In order to construct a PUF, a number of ROs is placed on an FPGA. To extract one bit, a pair of oscillators a and b is selected and their frequencies f_a and f_b are compared. If $f_a > f_b$, the PUF outputs a one, otherwise it outputs a zero. To extract a sequence of bits, for each bit a pair of ROs is compared.

The quality of a PUF construction can be evaluated by several measures. The relative Hamming distance (HD) of two responses from different devices (response inter-distance) should be close to 50% to fulfill uniqueness. To allow reproducibility, the relative HD of two responses from the same PUF (response intra-distance) should be close to 0%. A study in [5] confirmed that ROPUFs can exhibit useful properties. At normal operating conditions, the average response inter-distance is 47.31%, the average response intra-distance is 0.86%. Hence, ROPUFs fulfill both uniqueness and reproducibility.

[5] provides a large data set of real world RO frequencies. 193 FPGA chips (90nm) were used, each equipped with 512 ROs. To gather data, the authors evaluated each RO 100 times. The result is a data set with 9.881.600 entries which was obtained at normal operating conditions (1.2V, 25°C). Also data of 5 PUFs evaluated with different temperature and supply voltage conditions are available. We use the data set to verify the quality of the approach provided in [6], since we expect real world data to yield a more realistic model compared to theoretical assumptions.

We use the data set and implement a software framework, such that we are able to derive bit sequences from the provided frequencies. As in [5], we use the 512 ROs to extract a 511 bit sequence by comparing the frequencies of adjacent ROs. Since there exist 100 measurements per RO we extract the bit sequences 100 times and derive reliability information for each bit position. Reliability values are transformed into soft information which is used as input for the decoder.

3 Error Correction for Key Reproduction

Convolutional Codes, introduced in [1], recently were applied to error correction in PUFs [2,6]. They are adequate to the PUF scenario since efficient hardware implementations are possible, e.g. an area-optimized implementation of the Viterbi decoder for FPGAs is provided in [3]. The main idea of convolutional codes is to map length-k information blocks to length-n code blocks that depend also on the μ previous information blocks. Encoders can be implemented using linear shift registers with k inputs, n outputs and a memory of length μ. Efficient maximum-likelihood decoding is possible, which can be performed using the Viterbi algorithm [10]. The complexity of the Viterbi algorithm increases exponentially with μ, contrarily the error probability of the decoder decreases with μ. For details about convolutional codes we refer to [4, Chapter 8].

In order to reproduce keys, so-called *Secure Sketches* which use an error

correcting code and additional helper data are applied. The Code-Offset construction [8] is one of the most often used schemes. In an *initialization phase* helper data are obtained from an initial PUF response $\mathbf{r}_I$. Therefore, a random codeword $\mathbf{c}$ of a specified code $\mathcal{C}$ is chosen and added to $\mathbf{r}_I$, which results in the helper data $\mathbf{h}$. This phase occurs once in a secure environment during manufacturing. The *reproduction phase* occurs whenever the key has to be reproduced. Let $\mathbf{r}$ be the re-extracted key, which differs slightly from $\mathbf{r}_I$ due to the fuzzy nature of PUF responses and hence can be described as $\mathbf{r} = \mathbf{r}_I + \mathbf{e}$. Adding $\mathbf{h}$ to $\mathbf{r}$, we get $\mathbf{r} = \mathbf{c} + \mathbf{e}$. Since $\mathbf{e}$ is sparse, $\mathbf{r}$ can be decoded in order to get $\hat{\mathbf{c}}$. If the distance between $\mathbf{r}_I$ and $\mathbf{r}$ is within the error correction capability of $\mathcal{C}$, we have $\mathbf{c} = \hat{\mathbf{c}}$. Adding $\mathbf{h}$ to $\mathbf{c}$ recovers $\mathbf{r}_I$.

4 Results

To verify that the usage of convolutional codes is reasonable not only in simulations based on a mathematical model but also for real world PUFs, we apply the codes suggested in [6] to all the 193 ROPUFs provided by [5].

4.1 Normal Operating Conditions

The results in this section were obtained by using data collected at normal operating conditions (1.2V supply voltage and 25°C ambient temperature). The relative response intra-distance of the 193 PUFs in the data set is in the range between 0.38% and 1.39% [5], hence we have very good channels during stable operating conditions. For our test we first used convolutional codes of rate $R = \frac{1}{2}$. Using memory length $\mu = 10$, we were able to reconstruct all samples from the given data set using hard decision (hd) decoding. Providing soft information (sd) to the input of Viterbi, a code with memory length $\mu = 2$ is sufficient to reproduce all responses correctly. Decreasing the rate to $R = \frac{1}{3}$ allows to decrease the memory length to $\mu = 3$ in order to obtain good results providing only hard information. Table 1 summarizes the portion of the 193 PUFs for which at least one of 100 key reproductions fails in average when simulating 1000 times.

	$\mu = 2$ (hd)	$\mu = 3$ (hd)	$\mu = 6$ (hd)	$\mu = 10$ (hd)	$\mu = 2$ (sd)
$R = \frac{1}{2}$	33	15	1	0	0
$R = \frac{1}{3}$	1	0	0	0	0

Table 1: Number of PUFs for which key regeneration fails at least once.

4.2 Varying Operating Conditions

The data set contains 5 PUFs which were evaluated at 9 different temperature and supply voltage conditions. For each supply voltage and temperature combination 100 measurements were taken. Table 2 shows the average portion of failures when reproducing the key 100 times for the 5 PUFs using hard input with a $R = \frac{1}{2}$ convolutional code and memory length $\mu = 6$. For calculating the average 1000 runs were performed. Our experiments confirm the studies in [5]: Differences in temperature do not much increase the amount of errors. Lowering or rising the supply voltage however induces more errors. For small variances in temperature, the chosen code with hard input is sufficient.

Using instead $R = \frac{1}{3}$ and memory length $\mu = 6$ without soft information at the input of Viterbi, we are able to decode all voltage temperature combinations, except the 0.96V case. Providing soft information helps to circumvent problems with varying environment conditions. Even with a rate $\frac{1}{2}$ code having memory length $\mu = 2$, we were able to correctly decode all test cases in all runs.

	PUF 1	PUF 2	PUF 3	PUF 4	PUF 5
0.96V, 25°C	100	100	79	100	100
1.08V, 25°C	36	18	2	26	58
1.20V, 25°C	0	0	0	0	0
1.20V, 35°C	0	0	0	0	0
1.20V, 45°C	0	0	0	0	0
1.20V, 55°C	0	0	0	0	0
1.20V, 65°C	0	0	0	0	0
1.32V, 25°C	6	52	0	0	47
1.44V, 25°C	59	99	54	91	96

Table 2: Average amount of failures when regenerating the key 100 times under varying operating conditions using hard input ($R = \frac{1}{2}$, $\mu = 6$).

4.3 Larger Number of Synthetic Measurements

The main drawback of the experiments discussed above is the low number of available responses. Now we aim for a ROPUF model which is calibrated using real measurements in order to generate a large number of synthetic measurements. The model in this section uses a composition of frequencies for RO i as derived in [9], when aging of chips was studied.

$$f_i = f_{avg} + f_{PV} + f_{aging} + f_{noise}, \tag{1}$$

where the frequency f_i of RO i consists of f_{avg} (average delay, static component), f_{PV} (deviation due to process variation, static component), f_{aging}

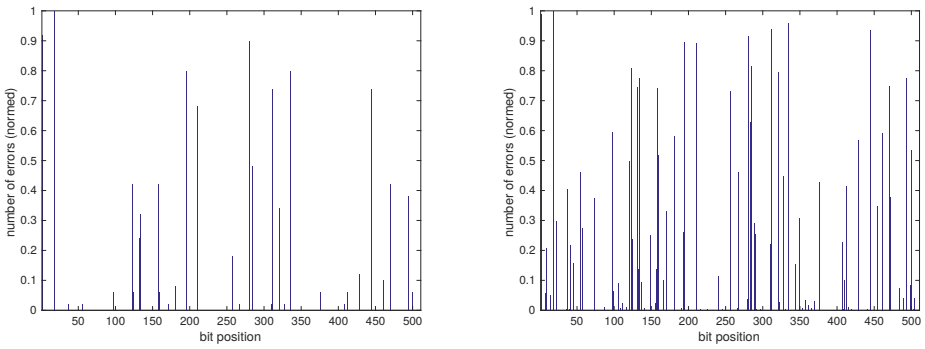

Figure 1: Comparison of error pattern between real data and synthetic data.

(dynamic component due to aging), and f_{noise} (dynamic component due to noise). For each RO we calculate the average of 100 readouts in order to remove noise and hence get the average and PV components. We model new responses by simulating $f_{aging} \sim N(-6.75, 0.05)$ and $f_{noise} \sim N(0, \sigma)$ where $\sigma = \sqrt{\frac{1}{99}\sum_{k=1}^{100}(f_k - f_{avg})^2}$. These distributions where derived in [9], by performing accelerated aging via stressing the FPGAs with temperatures up to 80°C and voltages up to 1.8V. Figure 1 shows for an exemplary PUF, that the synthetic measurements preserve the general error pattern of the real world data. Hence the simulation results of the synthetic data provide a good benchmark which encompasses the results obtained using the limited number of measurements in Sections 4.1 and 4.2. Table 3 shows the results of simulations using 10^7 synthetic responses generated using the model discussed in this section.

	$\mu = 2$	$\mu = 3$	$\mu = 6$	$\mu = 10$
$R = \frac{1}{2}$ (hd)	1.33e-01	4.99e-02	1.99e-03	2.19e-06
$R = \frac{1}{2}$ (sd)	3.82e-02	8.89e-04	2.70e-06	0
$R = \frac{1}{3}$ (hd)	4.28e-03	2.78e-03	0	0
$R = \frac{1}{3}$ (sd)	5.44e-05	0	0	0

Table 3: Word error probabilities when using synthetic responses.

5 Conclusion

Using $p \approx 0.15$ as usual done in the PUF community seems to be too pessimistic for ROPUFs, since we only obtain this order for extreme supply voltage conditions [5]. Under normal conditions, a $R = \frac{1}{2}$ convolutional code with memory

length $\mu = 2$ is sufficient when using soft information at the decoder's input. Without soft information, μ should be increased to 10. Another option is to decrease R to $\frac{1}{3}$, then $\mu = 3$ is sufficient. Dealing with varying environments, using soft information input we were also able to decode all test cases correctly (even the extremes) with a convolutional code of rate $\frac{1}{2}$ and $\mu = 2$. For approximations using synthetic responses we increase μ to 6. For implementation, the Viterbi decoder from [3] can be chosen, which is area and power-optimized for FPGAs. In total we observe that μ can be much smaller for 90nm ROPUFs than expected from the results in [2, 6]. Since p is much lower than assumed by most theoretical models, list decoding and multiple readouts used in [6], are not necessary for the specific ROPUFs considered in this work.

References

[1] P. Elias, Coding for Noisy Channels, in *Proc. of the Institute of Radio Engineers, vol. 43, no.3*, 1955, 356–356.

[2] M. Hiller, A.G. Önalan, G. Sigl and M. Bossert, Online Reliability Testing for PUF Key Derivation, *TrustED*, 2016, 15–22

[3] M. Hiller, L. Rodrigues Lima and G. Sigl, Seesaw: An Area-Optimized FPGA Viterbi Decoder for PUFs, *Digital System Design*, 2014, 387–393.

[4] M. Bossert, *Channel Coding for Telecommunications*, Wiley & Sons, 1999.

[5] A. Maiti, J. Casarona, L. McHale and P. Schaumont, A Large Scale Characterization of RO-PUF, in *IEEE Int. Symposium on Hardware-Oriented Security and Trust*, 2010, 94–99.

[6] S. Müelich and M. Bossert, Using Convolutional Codes for Key Extraction in Physical Unclonable Functions, in *Preprint arXiv:1704.01306*, 2017.

[7] G.E. Suh, S. Devadas, Physical Unclonable Functions for Device Authentication and Secret Key Generation, *Design Automation Conference*, 2007.

[8] Y. Dodis, L. Reyzin and A. Smith, Fuzzy Extractors: How to Generate Strong Keys from Biometrics and other Noisy Data, *Advances in Cryptology-Eurocrypt*, 2004, 523–540.

[9] A. Maiti, L. McDougall and P. Schaumont, The Impact of Aging on an FPGA-based Physical Unclonable Function, *Field Programmable Logic and Applications*, 2011, 151–156.

[10] A. Viterbi, Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm, *IEEE Transactions on Information Theory Vol.13*, 1967, 260–269.