Evaluation of a PUF Device Authentication Scheme on a Discrete 0.13um SRAM

Patrick Koeberl[1], Jiangtao Li[1], Roel Maes[2],
Anand Rajan[1], Claire Vishik[1], and Marcin Wójcik[3]

[1] Intel Corporation
{patrickx.koeberl,jiangtao.li,anand.rajan,claire.vishik}@intel.com
[2] Catholic University of Leuven
roel.maes@esat.kuleuven.be
[3] University of Bristol
wojcik@cs.bris.ac.uk

Abstract. The contamination of electronic component supply chains by counterfeit hardware devices is a serious and growing risk in today's globalized marketplace. Current best practice for detecting counterfeit semiconductors includes visual checking, electrical testing, and reliability testing, all of which require significant investments in expertise, equipment, and time. In TRUST'11, Koeberl, Li, Rajan, Vishik, and Wu proposed a new device authentication scheme using SRAM Physically Unclonable Functions (PUFs) for semiconductor anti-counterfeiting. Their authentication scheme is simple, low cost, and practical. However, the method and corresponding parameters of their scheme are based on a theoretical SRAM PUF model without support from real experimental data. In this paper, we evaluate a real SRAM PUF on a discrete 0.13um SRAM, and use the PUF result to evaluate this device authentication scheme and show that this scheme indeed works well. We identify several gaps between the theoretical model and the experimental SRAM PUF result, and adjust the parameters of the scheme accordingly. In addition, we provide a new post-processing function that results in a smaller false rejection rate and false acceptance rate.

Keywords: physically unclonable functions, device authentication, hardware security, anti-counterfeiting, implementation and evaluation.

1 Introduction

Semiconductor counterfeiting is a growing problem in today's globalized marketplace. The majority of counterfeit semiconductors detected today are remarked devices where a device's markings are forged in order to misrepresent aspects of the device's performance, brand or some other key specification. Such devices, if embedded in an electronic system may fail in the field when subjected to a different operational environment than the part was designed for. The consequences of such failures might range from minor inconvenience to the end user to loss of life for devices which are embedded in safety-critical infrastructure.

L. Chen, M. Yung, and L. Zhu (Eds.): INTRUST 2011, LNCS 7222, pp. 271–288, 2012.

A number of high-profile instances of counterfeit product entering the semiconductor supply chain have been reported, in one instance involving the US Air Force, microprocessors for its F-15 flight control computer were procured from a broker and found to have been remarked [12].

Current approaches to detecting semiconductor counterfeits range from non-destructive optical and x-ray inspection of device samples to destructive testing. Such practices require significant investments in time and expertise and in many cases can only be applied to a sample of the device population. Device traceability and authentication standards which can support an anti-counterfeiting strategy are beginning to emerge. For example, SEMI T20-1109 [14] defines standardized device traceability and authentication mechanisms based on encrypted serial numbers applied at a variety of package levels ranging from the device package itself to higher levels such as product and shipping packaging. An authentication service provides for validation of the serial numbers. It is conceivable that such standards could be applied at the silicon level, for example by programming the serial number into non-volatile memory (NVM) such as EEPROM, flash, or fuses. However, secure serialization mechanisms have the shortcoming that they are clonable by any competent counterfeiter.

An alternative approach is to utilize the intrinsic properties of the silicon to enable a class of identification and authentication applications. Physically Uncloneable Functions (PUFs) are a promising security primitive that exploit the manufacturing variation inherent in any mass produced object to derive biometric-like fingerprints which are difficult to clone, even for the manufacturer. PUFs which exploit the process variation inherent in Integrated Circuit (IC) manufacturing are of particular interest due to the high levels of integration achievable in modern CMOS technologies.

Recently Koeberl, Li, Rajan, Vishik, and Wu proposed a new device authentication scheme using SRAM PUFs for semiconductor anti-counterfeiting [8]. In their scheme, each device is embedded with a small SRAM PUF which serves as an intrinsic unclonable fingerprint of the device. At manufacturing time, the manufacturer evaluates the PUF and extracts the m-bit PUF result into a short k-bit device ID. The manufacturer then creates a device certificate based on the device ID. Any verifier can authenticate the device by evaluating the SRAM PUF, re-computing the device ID, and verifying the device certificate. This scheme is simple and practical as it does not require any online databases or on-chip cryptographic operations. For hardware devices which already have SRAM and non-volatile storage embedded, this scheme takes almost no additional cost.

The security of the device authentication scheme [8] relies on the size of m, the size of the SRAM PUF. They assume that it is too expensive or uneconomical for an adversary to embed an m-bit PUF simulator into the non-volatile memory or circuit of a counterfeit device. This assumption is reasonable for economically motivated attackers and integrated circuits implemented in modern technology nodes. It is important to keep m reasonably large, while keeping k small to reduce the size of device certificate. The paper [8] provided a post-processing function to compress the m-bit PUF result into a k-bit device ID using a theoretical SRAM PUF model.

1.1 Our Contribution

Our paper can be seen as an improvement to [8] with the following contributions.

- We implement the device authentication scheme using a discrete 0.13μm SRAM chip as the SRAM PUF and show that the authentication scheme works well. We also show that the post-processing function in [8] is reasonably effective, compressing a 256-kb PUF into a 512-bit device ID with both False Reject Rate (FRR) and False Acceptance Rate (FAR) under 10^{-10}.
- Although the evaluated SRAM PUF exhibits low levels of bias ($< 1\%$) we discover that the PUF response is highly correlated with an estimated entropy of 63% or less. We consider this to be an important result since other work in the literature on SRAM PUFs assumes that the SRAM cell power-up states are independently distributed. This assumption may be incorrect for particular SRAM instantiations.
- We provide a couple of improvements of the device authentication scheme. One is that we modify the device certificate to address the device remarking issues. Second, we provide a new post-processing function which is more effective when the SRAM PUF result is biased or correlated. We show that our post-processing function can compress a 256-kb PUF into a 512-bit device ID with both FRR and FAR under 10^{-13}.

1.2 Related Work

Device authentication protocols typically rely on the secure storage of a cryptographic secret in non-volatile on-chip memory such as EEPROM, flash or fuses. Cloning of the device by extracting the secret and replicating it in another device instance is a possibility, unless explicit steps are taken to protect the secret. For example, the Trusted Platform Module (TPM) [16] uses a protected private key in non-volatile memory to enable remote device authentication and attestation applications. The approach taken in TPM may not be suitable for detecting semiconductor counterfeits.

In 2007, Suh and Devadas proposed a low cost authentication scheme based on silicon PUFs and using a challenge response protocol [15]. This authentication scheme places a number of constraints on the silicon PUF, which must posses a large number of challenge-response pairs, and the system since authentications must be on-line. In this paper, we choose to implement and evaluate the offline authentication scheme [8] instead, as we believe the offline authentication scheme has few limitations and is more appealing to the real applications.

An SRAM fingerprinting method is proposed in [6], where the power-up state of SRAM cells is used in a device identification scheme. Experiments show that a 64-bit SRAM fingerprint is sufficient to uniquely identify devices among a small population of 5,120 instances. A key difference between this work and the ideas in [8] is that the scheme's resistance to cloning attacks is not a design criterion.

Another related work is the authentication scheme in [3], which provides a strong binding between the paper medium and the data on it using a fingerprint extracted from the ultraviolet fibers. This scheme can be used to detect counterfeited tickets, banknotes, and prescriptions. The device authentication scheme in [8] shares some similarities between this scheme, however, it is different in that [8] is optimized for anti-counterfeiting of electronic devices and uses a silicon PUF from the hardware device.

1.3 Paper Outline

The rest of the paper is organized in the following way. We first review the concept and constructions of PUF in Section 2. We then review the device authentication scheme of [8] in Section 3 and provide our improvements. We outline our experimental setup and evaluation methodologies in Section 4. The results of the evaluations are analyzed in Section 5. We conclude our paper and discuss future work in Section 6.

2 Physically Unclonable Functions

Physically Unclonable Functions are physical challenge-response systems which when challenged respond with unique and unpredictable responses. PUFs are also *physically unclonable*, in other words it is extremely difficult to create a physical copy of a PUF with the same challenge-response behaviour as the original. Physical unclonability is achieved in all known PUFs by deriving the PUF response from the manufacturing variation inherent in any mass produced object. The PUF concept was introduced in [13] where the random arrangement of scattering particles in a transparent medium is the basis of an optical PUF. Silicon PUFs, introduced in [4], exploit the manufacturing variation inherent in the CMOS fabrication process. Variations in physical parameters such as transistor dopant concentrations and line widths result in measurable differences in circuit delays. Silicon PUFs are of considerable interest as they can leverage the high levels of integration possible in modern CMOS technology nodes.

A silicon PUF embodiment based on SRAM was introduced in [5]. Here, the power-up state of SRAM cells is used as the PUF response. A typical six-transistor SRAM cell is shown in Figure 1. The storage element in an SRAM cell consists of four cross-coupled transistors, denoted in the figure as M1, M2, M3 and M4. The cross-coupled structure is bistable i.e. it can assume one of two stable states. The power-up state for a particular cell is determined by the relative characteristics of the transistors forming the cross-coupled structure. Mismatches due to manufacturing variation of the transistors will cause the cell to have a preference to power-up in a particular state, a phenomenon that can be exploited as a PUF.

It is useful to consider SRAM PUFs as members of a larger grouping which we term cross-coupled PUFs due to the cross-coupled structure forming the bistable storage element. In fact, any digital storage element constructed from static logic will use a cross-coupled structure as its basis and one can envisage cross-coupled

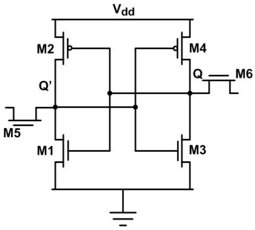

Fig. 1. Construction of an SRAM cell

PUFs based on the many flip-flop and latch variants available to the digital designer. An example of a cross-coupled PUF based on D-type flip-flops can be found in [9].

3 Device Authentication with SRAM PUFs

In this section, we first review the off-line device authentication scheme presented in [8] and then provide two improvements of this scheme.

3.1 Review of Off-Line Authentication Scheme

We now review the off-line device authentication scheme in [8] as follows. This scheme has two main building blocks: a digital signature scheme [11] and a family of SRAM PUFs. A digital signature scheme requires par of public key for device manufacturer's verification and private key for signing. For our applications we can divide this off-line authentication scheme on two phases: an enrolment phase Figure 2 and an evaluation phase Figure 3. In the former, the manufacturer certifies each device and ships them into the market; in the latter, the verifier accepts or rejects the hardware device after applying the verification procedure.

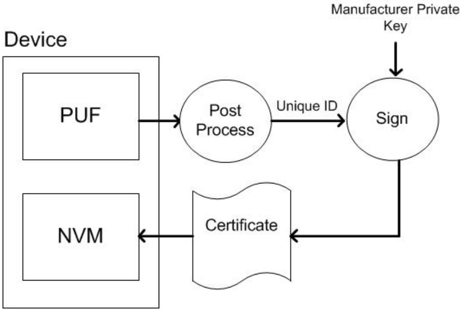

Fig. 2. Enrolment phase of the off-line device authentication scheme

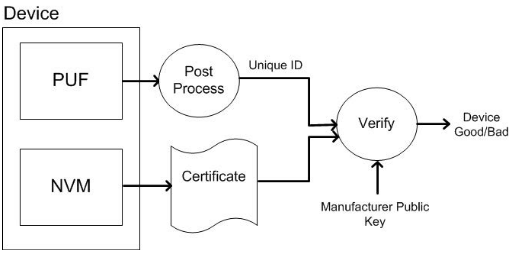

Fig. 3. Evaluation phase of the off-line device authentication scheme

Having those above-mentioned assumptions we can describe the off-line authentication scheme as follows:

Enrolment Phase. In this phase the manufacturer instantiates an SRAM PUF into the device D and runs the evaluation procedure to obtain the unique identity s. In the next step the manufacturer computes the device ID id_D using a post-processing function and creates a signature σ of the ID using private key. The last step of this procedure is to store previously generated signature and unique device ID as the device's certificate in the NVM of the device.

Evaluation Phase. In this phase the verifier who wants to verify the device runs the evaluation procedure of the SRAM PUF in the device and obtains s'. Having s', the verifier uses the post-processing function and obtains id'_D. The verifier then reads the certificate stored in the NVM of the device and uses the public key to verify the signature σ on id_D. If this step fails, the device is rejected otherwise the verifier checks the Hamming distance between id_D and id'_D. If it is greater than the previously set security parameter δ the device is rejected, otherwise verifier accepts the device.

Both the enrolment and evaluation phases use the post-processing functions to map an m-bit string to a k-bit string. The security of the device authentication scheme [8] relies on the value of m. They assume that it is too expensive or uneconomical for an adversary to embed an m-bit PUF simulator into the non-volatile memory or circuit of a counterfeit device. Thus it is important to keep m reasonably large, while keeping k small to reduce the size of device certificate. Observe that standard hash functions are not noise preserving, i.e., one small difference in the input leads to a large difference in the output, and thus we can not use them in our application. An efficient post-processing function is introduced in [8] and analyzed based on a theoretical SRAM PUF model where each PUF cell is independently and randomly distributed with small noise. We denote this post-processing function as $f_1 : \{0,1\}^m \to \{0,1\}^k$. This function can be computed in the following three steps:

1. Let ℓ be the largest odd number such that $k \cdot \ell \leq m$.
2. Divide the first $k \cdot \ell$ bits of string s into k groups $G_1, \ldots, G_k$, where each group has ℓ bits. The mapping from bits in s to k groups is random but fixed per function and is encoded in the algorithm.
3. For each group G_i, where $1 \leq i \leq k$, compute $t_i = \text{Voting}(G_i)$, the majority voting result of bits in G_i. More specifically, let $G = \{b_1, \ldots, b_\ell\}$ where $b_1, \ldots, b_\ell \in \{0, 1\}$. The majority voting function Voting(G) is defined as follows: Voting(G) outputs 1 if $b_1 + \cdots + b_\ell > \ell/2$ and outputs 0 otherwise.
4. The final output of f_1 is $t_1, t_2, \ldots, t_k$.

As in [8], we use the following terms to analyze the effectiveness of the post-processing functions.

Definition 1 (False Rejection Rate). *If the manufacturer certifies a legitimate device in the enrollment phase, the False Rejection Rate (FRR) is the probability that the device fails to be verified in the evaluation phase.*

Definition 2 (False Acceptance Rate). *The False Acceptance Rate (FAR) is the probability that an uncertified device with a random SRAM PUF embedded can be successfully verified in the evaluation phase, assuming the attacker can inject a valid device certificate into the counterfeit device.*

3.2 Our Improvements

We give two improvements to the device authentication scheme. The first is a new post-processing function which is more effective when the SRAM PUF result is biased. The second improvement is that we include additional data in the device certificate to address issues related to device remarking attacks.

In [8], the post-processing function f_1 is based on a theoretical model in which each SRAM PUF bit is randomly and independently distributed. In practice, a small bias in the SRAM PUF could exist. Some proposed SRAM architectures may exhibit larger biases due to specific features such as asymmetric designs intended to address leakage power and read stability in recent technology nodes [7,2]. As shown in Table 1, even a small bias in the raw PUF response will be significantly amplified in the device ID after the majority voting. As a result, the inter-distance and entropy of the device IDs may be significantly reduced. Small inter-distances will result in an increase in the FAR.

Table 1. Probability of '0' in Device ID after majority voting

Probability of '0' in PUF response	50%	50.5%	51%	52%	55%
Group size = 255	50%	56.35%	62.54%	73.88%	94.55%
Group size = 511	50%	58.95%	67.45%	81.73%	98.83%
Group size = 1023	50%	62.55%	73.89%	89.98%	99.93%

The motivation for a new post-processing function is to minimize the effect of a slight bias to '0' or '1' in the SRAM PUF response. Our method is straightforward, we first apply XOR to the PUF response to remove bias, and then perform the majority voting. Note that, applying XOR to the PUF result will also increase the noise rate in the device ID. As shown in Table 2, assuming each bit in the PUF response is independently distributed, the bias in the device ID reduces significantly after we perform bit-wise XOR on the PUF response.

Table 2. Probability of '0' in Device ID after XOR and majority voting

Probability of '0' in PUF response	50%	50.5%	51%	52%	55%
Probability of '0' after bitwise XOR	50%	50%	50.02%	50.08%	50.5%
Group size = 255 after XOR	50%	50%	50.26%	51.02%	56.35%
Group size = 511 after XOR	50%	50%	50.36%	51.44%	58.95%
Group size = 1023 after XOR	50%	50%	50.51%	52.04%	62.55%

A new post-processing function. We now introduce a new post-processing function, denoted as f_2, as a generalization of the one in [8] but designed especially to remove any bias in the PUF data using an XOR operation. Function f_2 can be computed in the following five steps:

1. Let d be a small integer, a parameter to this function.
2. Let ℓ be the largest odd number such that $k \cdot \ell \cdot d \leq m$.
3. Divide the first $k \cdot \ell \cdot d$ bits of string s into k groups $G_1, \ldots, G_k$, where each group has $\ell \cdot d$ bits. The mapping from bits in s to k groups is random but fixed per function and is encoded in the algorithm.
4. For each group G_i, where $1 \leq i \leq k$, compress $\ell \cdot d$ bits into an ℓ-bit group G'_i using the XOR operation as follows. Let $G = \{b_0, b_1, \cdots, b_{\ell \cdot d-1}\}$. $G' = \{c_0, c_1, \cdots, c_{\ell-1}\}$ is computed by setting $c_j = b_{d \cdot j} \oplus b_{d \cdot j+1} \oplus \cdots \oplus b_{d \cdot j+d-1}$, for $j = 0, \ldots, \ell - 1$.
5. For each group G'_i, where $1 \leq i \leq k$, $t_i = \text{Voting}(G'_i)$, the majority voting result of bits in G'_i. The final output of f_2 is $t_1, t_2, \ldots, t_k$.

Note that the function f_2 is similar to the function f_1, except that f_2 reduces any bias using the XOR operation [17]. The function f_1 is a special case of the function f_2 with parameter $d = 1$ and those functions can be treated as a family of post-processing functions. Nevertheless, we analyze them separately to stress that the first one will not reduce any bias. The XOR operation will also remove any correlations in the SRAM PUF response. In Section 4.2 we show that although the bias of the SRAM PUF response is small, it is found to be highly correlated. We shall show in Section 5 that the function f_2 is indeed better than f_1 for correlated, rather than biased SRAM PUF responses.

Configuration data in device certificate. The above device authentication scheme binds the device certificate with the device ID computed from the embedded PUF. Observe that this scheme only proves a hardware device is a legitimate device certified by the manufacturer but it does not address the device remarking attack, in which the attacker buys a legitimate low-end device from a device manufacturer and remarks it as a high-end device from the same manufacturer.

We can easily address this attack by adding configuration data in the data certificate signed by the manufacturer private key. The configuration data contains additional information about the device, such as model number, speed grade, size of NVM, size of SRAM, and device features. In the evaluation phase, the verifier validates not only the device ID and the signature, but also the configuration data in the certificate. This effectively addresses the remarking attack, unless that attacker can break the signature scheme or clone a PUF.

4 Experimental Methodology

In this section, we present the methodology used to evaluate the SRAM PUF performance and discuss the results in terms of PUF characteristics. The authentication scheme performance based on these results are given in Section 5.

4.1 PUF Performance

The following methodology was used to evaluate SRAM PUF performance. The experimental data is limited to a single 1MB Zero Bus Turnaround (ZBT) SRAM chip manufactured by ISSI on a 0.13μm CMOS process. Measurements were obtained at room temperature and nominal supply voltages. Note that SRAM PUF noise rates are influenced by the voltage and temperature operating conditions. In [5], temperature ranges of -20°C to 80°C are reported to result in maximum fractional hamming distances of 12% when compared to a reference measurement at 20°C. In the PUF based device authentication scheme in Section 3, the enrolment and evaluation processes both occur in production environments where temperature is controlled. Device supply voltages are typically controlled to within ± 5% or better either by the device tester or similar during enrolment and by the device power supply subsystem during evaluation. We therefore consider it reasonable to perform SRAM PUF measurements at room temperature and at nominal supply voltages. For further details on the experimental setup please consult the Appendix.

To emulate multiple PUFs on a single physical SRAM, the 1MB SRAM address space was divided into 32 logical PUFs of 32kB each. Inter- and intra-distance measures are used to evaluate the effectiveness of the 32 logical SRAM PUFs. The *inter-distance* metric measures the Hamming distance between two measurements (responses) collected from different (logical) PUF instances. Inter-distance assesses the uniqueness of a PUF response and ideally should be close to half the response length. The *intra-distance* metric measures the Hamming distance between responses collected from a single logical PUF instance at different moments. Intra-distance assesses the (un)reliability of a PUF response and ideally should be close to zero. The usability of a particular PUF implementation can be quickly evaluated by looking at the separation between its inter- and

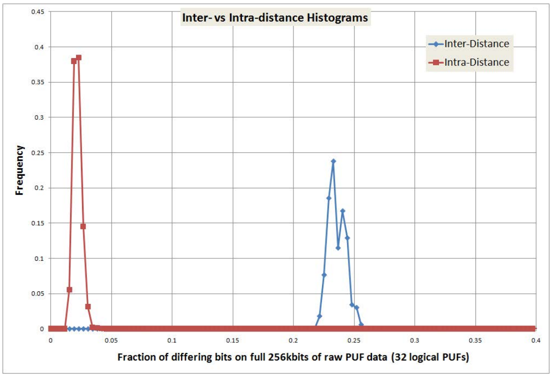

Fig. 4. Inter- vs intra-distance histograms

intra-distances. An implementation is said to show a good PUF behavior if on average its inter-distances are much larger than its intra-distances.

We evaluate the PUF behavior of the observed SRAM dumps. For every of the 32 logical PUFs, one of the 100 dumps is selected as a reference measurement. Intra-distances are calculated by comparing the remaining 99 dumps of every logical PUF to its respective reference measurement and counting the number of differing bits. Inter-distances are calculated by comparing the reference measurements of every possible pair of logical PUFs and counting the number of differing bits. The occurrence of inter- and intra-distances in our data set is summarized as a histogram in Figure 4, with inter- and intra-distances expressed as a fraction of the full logical PUF size of 32kB on the X-axis. This histogram shows that in our experiment the observed intra-distance is on average $\mu_{intra} = 2.2\%$ of the measured response length, which is in line with the results in [5] and is considered reasonable for measurements obtained at room temperature. The average inter-distance of our measured responses is around $\mu_{inter} = 23.6\%$ of the response length. This sub-optimal average inter-distance result is indicative of some level of bias in and/or correlation between (logical) PUF instances and will be further explored in Section 4.2. However, the observation that $\mu_{inter} >> \mu_{intra}$ is a strong indication that the uninitialized power-up values of the considered SRAM memory show good PUF behavior.

4.2 Bias and Correlation

Ideally one would expect the average inter-distance to be 50% of the response length when all the response bits are unbiased and independent. Any statistically significant deviation from 50% indicates either a bias in the bit values, a dependence between different bit values, or both. Since we observe an average inter-distance of $23.6\% < 50\%$ we investigate the cause.

To evaluate a possible bias we consider the number of observed 1-values in the reference measurements of all 32 logical PUFs. The smallest number of observed 1-values is 128458 (49.00% of a 32kB PUF) and the largest number is 129737 (49.49% of a 32kB PUF). Although these values are very close to 50%, there is still a statistically significant deviation because the sample set is large. The respective observed p-values for an hypothesis of unbiased bits are $1.8 \cdot 10^{24}$ and $1.9 \cdot 10^{-7}$ which are a strong indication to reject this hypothesis and assume there is a bias. However, the observed bias is too small ($< 1\%$) to be the only cause for the small inter-distances.

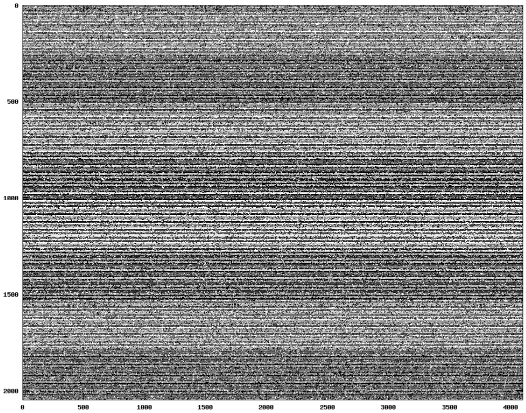

Fig. 5. Single dump of 1 MB SRAM

In order to investigate dependencies between different bits, we plot a single dump of the 1MB SRAM memory as a 2048x4096 bitmap, with a white pixel indicating a power-up value equal to 1 and a black pixel a power-up value of 0 for the considered bit. This bit map is shown in Figure 5 and an enlarged portion of this figure is shown in Figure 6. It is immediately clear from the observed patterns in these bitmaps that there exists a strong location-based correlation in the SRAM dump. From the enlarged plot, it is clear that consecutive lines have a strong tendency to power up with opposing values. From the full plot, additional large-scale patterns can be observed as darker and lighter bands in the bitmap. Similar patterns arise for any arrangement of the data where the number of lines and columns are a power of two. Since we defined logical PUFs as 32 ($= 2^5$) blocks of 262144 ($= 2^{18}$) consecutive bits from a single dump, strong correlations between different logical PUFs can be expected. This is the main cause for the observed small average inter-distance.

The underlying cause for these strong correlations is most likely to be found in the physical layout of the SRAM memory cells as a huge 2D array on the silicon die. In a typical SRAM architecture, cells in the same row and/or column

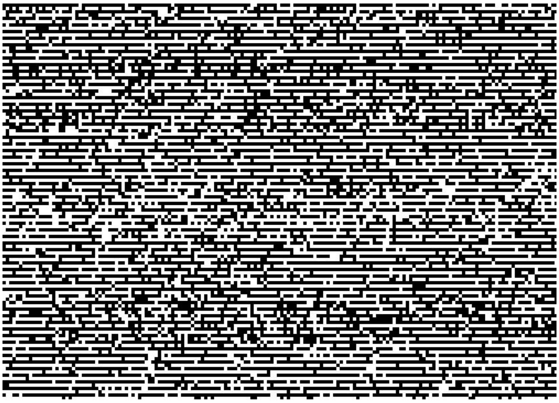

Fig. 6. Enlarged portion of a single SRAM dump

share a couple of elements. Cells in the same row are on the same word line, and cells in the same column share a couple of bitlines and a sense amplifier. A physical bias in the operation of any of these shared elements can cause a bias in all the cells connected to this element, which will show up as row- or column-based correlations in the PUF data, very similar to what we observe in our plots.

More generally, in addition to reducing the average inter-distance, these correlations will also severely decrease the expected entropy in the SRAM PUF response. Assessing entropy exactly is very hard, but an upper bound can be provided based on the compressibility of the data, since entropy is a lower bound for the smallest achievable compression. Using standard file compression techniques (zip), our 1MB SRAM dump files can be compressed to about 630kB, indicating an entropy level of 63% or less. We consider this an important result since such strong correlations leading to severely reduced entropy levels were never observed before for similar SRAM PUF constructions. In fact, many other works on SRAM PUFs or SRAM fingerprinting present very high estimated entropy levels of $> 90\%$ or assume an independent distribution of SRAM power-up states [1,10,6]. It is important to emphasize that although we observe correlations between different logical PUFs on the same device, the finding is of importance for the typical case where each device instantiates a single physical PUF. From our results it is clear that the actual entropy of an SRAM PUF will depend a lot on the physical instantiation of the SRAM memory and cannot be assumed to be very high without analysing its responses. Moreover, we show that merely looking at the bias in the responses is not sufficient, since strong dependencies between different bits can arise. For our data, the bias is very small ($< 1\%$) whilst we still observe severely reduced entropy levels ($< 63\%$).

5 Results and Analysis

The performance of our post-processing schemes is presented in this section. The key metric for our application is the FAR/FRR which we wish to maximise while keeping the storage cost of the device ID in bits as low as possible.

5.1 Result of the Function f_1

Figure 7 and Figure 8 show the results of applying the function f_1 for device IDs of 256- and 512-bits respectively. When compared to the raw PUF data of Figure 4, a degradation of the intra- and inter-distance results are observed, up around to a maximum of approximately 10% for the 512-bit device ID inter-distance result. The results show the first post-processing function to be largely noise preserving while also preserving the poor inter-distance results exhibited by the raw PUF data.

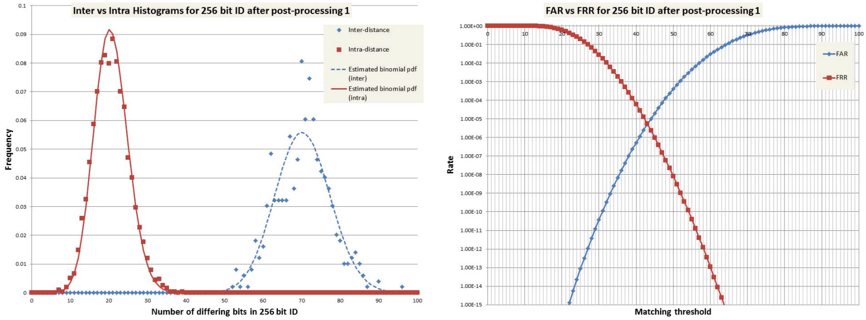

Fig. 7. Inter- vs intra-distance of 256-bit device IDs using f_1 and corresponding FAR/FRR rates

The FAR/FRR is estimated as follows. We model the inter-distance histogram as the probability density function for the bit difference between two device IDs; the FAR is the corresponding cumulative distribution function. Similarly, we model the intra-distance histogram as the probability density function for the number of error bits in the device ID; the FRR is the corresponding cumulative distribution function. The FAR/FRR for a 256-bit device ID is on the order of 10^{-5} which is unacceptable for most device authentication applications. Although the FAR/FRR performance of the 512-bit device ID is reasonable, at around 10^{-11}, the poor inter-distance result of the raw PUF data is preserved (there is a slight increase). In effect the low entropy of the raw PUF data is reflected in the resultant device ID. The efficiency of the 512-bit configuration is low as a result, although from the FAR/FRR perspective the performance is acceptable if the 512-bit device ID does not pose a storage issue.

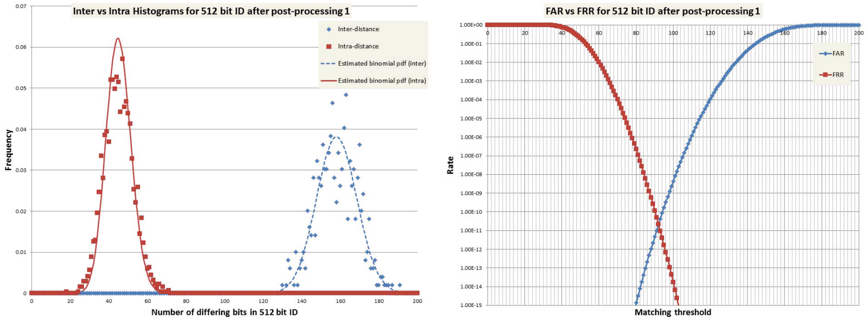

Fig. 8. Inter- vs intra-distance of 512-bit device IDs using f_1 and corresponding FAR/FRR rates

5.2 Results of the Function f_2

The results of the function f_2 for a 256-bit device ID with the XOR parameter $d = 2$ are shown in Figure 9. When compared to f_1 in the 256-bit configuration, an increase in the average noise rate as evidenced by the intra-distance result is observed, from approximately 8% to 14%. The inter-distance result shows a marked improvement to approximately 41% which approaches the 50% ideal. In terms of FAR/FRR the result is on the order of 10^{-7}, a result which is acceptable for authenticating reasonably large device populations. Observe that the XOR operation on the PUF output propagates PUF errors and increases the noise rate in the device ID. For a given threshold δ, the FRR becomes larger in f_2. However, note that the inter-distance increases as well after the XOR operation, the curve of FAR shifts to right. This allows us to choose a larger threshold δ for f_2 such that both FAR and FRR are smaller than using f_1.

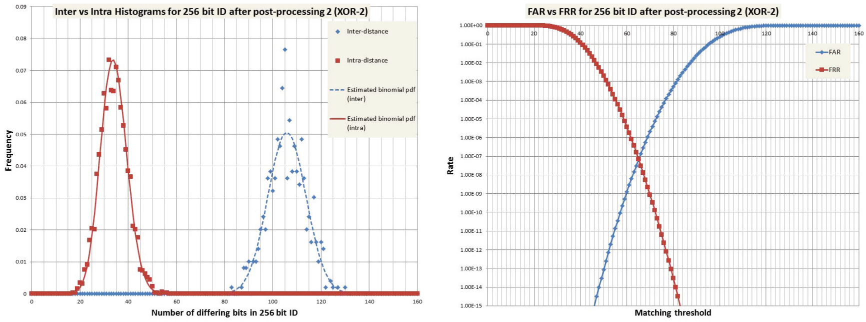

Fig. 9. Inter- vs intra-distance of 256-bit device IDs using f_2 with $d = 2$ and corresponding FAR/FRR rates

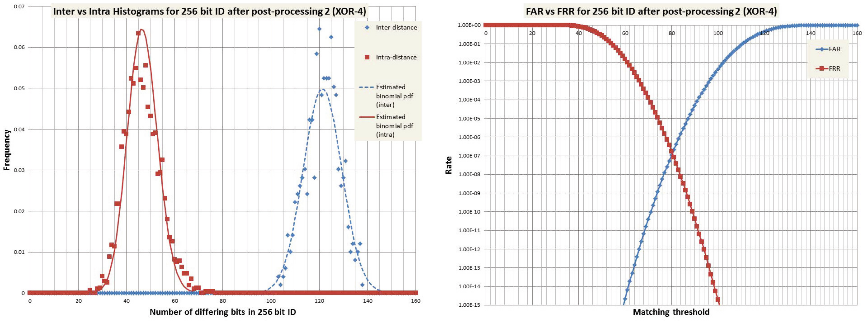

Fig. 10. Inter- vs intra-distance of 256-bit device IDs using f_2 with $d = 4$ and corresponding FAR/FRR rates

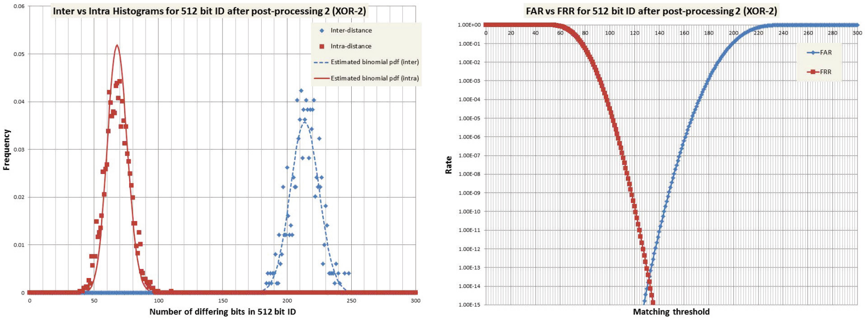

Fig. 11. Inter- vs intra-distance of 512-bit device IDs using f_2 with $d = 2$ and corresponding FAR/FRR rates

Figure 10 shows the results for a 256-bit device ID with $d = 4$. The intra-distance result indicates an average noise rate of more than double that of the first post-processing function. The inter-distance result is close to ideal at 47%. As for the $d = 2$ configuration above, the FAR/FRR of 10^{-7} may be acceptable for some applications.

In terms of FAR/FRR we see the best performance when using a 512-bit device ID with $d = 2$ as shown in Figure 11. In practice the rate of 10^{-13} can be considered negligible. As for the 256-bit, $d = 2$ case a similar increase in the average noise rate is observed as evidenced by the intra-distance result. Similarly, the inter-distance result approaches the 50% ideal.

5.3 Analysis

The considered post-processing functions affect the bias of the bit values in the output, and therefore also the average inter-distances. The effect of post-processing on the bias is shown in Table 3. From the analysis of SRAM PUF measurements in Section 4.2, it was clear that there exists a small ($< 1\%$) though statistically significant bias on the raw observed bit values. The theoretical treatment of the function f_1 in [8] predicts that the majority voting operation will deteriorate an existing bias in the raw data, and the measured results as shown in Table 3 support this claim. To overcome this issue, we introduced a second post-processing function f_2 which attempts to remove any bias prior to majority voting by XOR-ing a number of bits together. It is clear from Table 3 that even XOR-ing over a very small number of bits (2 to 4) removes the bias almost completely. In fact, the obtained results for f_2 show no statistically significant deviation from an unbiased source. As a direct consequence, the function f_2 produces much better FAR/FRR characteristics for the same ID length than function f_1.

Table 3. Average bias in the output bits after the different post-processing functions

	Raw PUF data	f_1	$f_{2,d=2}$	$f_{2,d=4}$
Full PUF dump (32kB)	49.21%	-	-	-
256 bit ID	-	30.35%	49.78%	49.73%
512 bit ID	-	35.85%	49.46%	50.42%

6 Conclusions

In this paper we presented the experimental results of a PUF device authentication scheme on a discrete 0.13μm SRAM. We evaluate the post-processing function presented in [8] and show that a 256-kb PUF can be compressed into a 512-bit device ID while maintaining an FAR and FRR of better than 10^{-10}. During the analysis it is observed that the SRAM PUF is strongly correlated with a small bias of less than 1%. A upper bound on the entropy level of the complete 1MB SRAM is estimated at 63%. We consider this to be an important result, since it implies that SRAM PUF entropy levels can be severely reduced even when the observed bias is small. Our results show that the entropy of an SRAM PUF can depend strongly on the SRAM PUF architecture and physical implementation.

We introduce a new post-processing function which shows good performance when presented with strongly correlated PUF responses such as we encounter in this paper. We show that this new function exhibits a negligible FAR and FRR when compressing a 256-kb PUF into a 512-bit device ID.

Future work will include a more detailed analysis of the SRAM PUF correlations observed in order to determine the root cause, and experimental evaluation of the device authentication scheme presented here on multiple physical SRAM instances. The robustness of the scheme to expected environmental swings will also be evaluated.

Acknowledgement. This work has been supported in part by the European Commission through the FP7 programme UNIQUE. We thank the anonymous reviewers for their helpful reviews and comments.

References

1. Armknecht, F., Maes, R., Sadeghi, A.-R., Sunar, B., Tuyls, P.: PUF-PRFs: A new tamper-resilient cryptographic primitive. In: Advances in Cryptology – EUROCRYPT 2009 Poster Session, pp. 96–102 (2000)
2. Azizi, N., Moshovos, A., Najm, F.N.: Low-leakage asymmetric-cell sram. In: Proceedings of the 2002 International Symposium on Low Power Electronics and Design, ISLPED 2002, pp. 48–51. ACM, New York (2002)
3. Bulens, P., Standaert, F.-X., Quisquater, J.-J.: How to strongly link data and its medium: the paper case. IET Information Security 4(3), 125–136 (2010)
4. Gassend, B., Clarke, D., van Dijk, M., Devadas, S.: Silicon physical random functions. In: ACM Conference on Computer and Communications Security, pp. 148–160. ACM Press, New York (2002)
5. Guajardo, J., Kumar, S.S., Schrijen, G.-J., Tuyls, P.: FPGA Intrinsic PUFs and Their Use for IP Protection. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 63–80. Springer, Heidelberg (2007)
6. Holcomb, D.E., Burleson, W.P., Fu, K.: Initial SRAM state as a fingerprint and source of true random numbers for RFID tags. In: Conference on RFID Security 2007, Malaga, Spain, July 11-13 (2007)
7. Kim, J.-J., Rao, R., Kim, K.: Technology-circuit co-design of asymmetric sram cells for read stability improvement. In: 2010 IEEE Custom Integrated Circuits Conference (CICC), pp. 1–4 (September 2010)
8. Koeberl, P., Li, J., Rajan, A., Vishik, C., Wu, W.: A Practical Device Authentication Scheme Using SRAM PUFs. In: McCune, J.M., Balacheff, B., Perrig, A., Sadeghi, A.-R., Sasse, A., Beres, Y. (eds.) Trust 2011. LNCS, vol. 6740, pp. 63–77. Springer, Heidelberg (2011)
9. Maes, R., Tuyls, P., Verbauwhede, I.: Intrinsic pufs from flip-flops on reconfigurable devices. In: 3rd Benelux Workshop on Information and System Security (WISSec 2008), Eindhoven, NL, p. 17 (2008)
10. Maes, R., Tuyls, P., Verbauwhede, I.: Soft decision helper data algorithm for sram pufs. In: Proceedings of the 2009 IEEE International Conference on Symposium on Information Theory, ISIT 2009, vol. 3, pp. 2101–2105. IEEE Press, Piscataway (2009)
11. Menezes, A., van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press (1996)
12. U. S. G. A. Office. Defense supplier base: Dod should leverage ongoing initiatives in developing its program to mitigate risk of counterfeit parts. GAO-10-389 (March 2010)
13. Pappu, R.S.: Physical one-way functions. PhD thesis, Massachusetts Institute of Technology (March 2001)
14. SEMI T20-1109. Specification for authentication of semiconductors and related products (2009), http://www.semi.org/
15. Suh, G.E., Devadas, S.: Physical unclonable functions for device authentication and secret key generation. In: Design Automation Conference, pp. 9–14. ACM Press, New York (2007)

16. Trusted Computing Group. TCG TPM specification 1.2 (2003), http://www.trustedcomputinggroup.org
17. von Neumann, J.: Various techniques used in connection with random digits. In: Householder, A.S., et al. (eds.) The Monte Carlo Method. National Bureau of Standards, Applied Mathematics Series, vol. 12, pp. 36–38 (1951)
18. Xilinx Inc. ML501 Evaluation Platform - User Guide, UG226 (v1.4), August 24 (2009)

A Experimental Setup

The experimental setup is based on the ML501 development platform from Xilinx [18] housing a Virtex-5 XC5VLX50-1FFG676 FPGA chip. Collecting SRAM PUF data directly from the FPGA chip is very difficult due to the automated initialisation procedure of the internal FPGA SRAM blocks, which is hard to circumvent. Instead, we selected the Zero Bus Turnaround (ZBT), high-speed, synchronous SRAM available on the board to collect experimental SRAM PUF data. This SRAM chip (IS61NLP25636A-200TQL) is manufactured using 0.13μm CMOS process technology by ISSI. The memory is organized as 256k x (32+4) bits (four parity bits, which are discarded in our case) which gives 1MB of total memory available for the analysis.

The development board is connected to the workstation via a serial null modem cable and the SRAM data is transmitted using the RS-232 standard. Python scripts and the library for serial connections are used to control the transmission on the workstation side. On the board side, the SRAM read-out is handled by an FPGA design containing a ZBT memory controller, a UART interface and a small data flow controller. A single complete readout of the 1MB SRAM memory takes about two minutes with the RS-232 baudrate set to 115200kbps. To read out a 32kB SRAM PUF, we estimate that it would take less than 4 seconds. After a complete memory measurement, the board is powered off and on again to collect the next SRAM dump. To assure a complete discharge of all on-board capacitors, a delay of at least 10 seconds is kept between two consecutive power cycles.

Using this measurement setup, 100 consecutive dumps of the 1MB uninitialized SRAM memory were collected and analyzed. All measurements were obtained at an ambient temperature around 293K (room temperature). Measurements obtained when the chip was cold, i.e., after a prolonged ($> 10s$) power-off time, were discarded. Further improvement of the measurement setup might include automatic control of the power cycling as well as increasing the data transmission speed.