Reverse Engineering and Prevention Techniques for Physical Unclonable Functions Using Side Channels

Sheng Wei* James B. Wendt* Ani Nahapetian†* Miodrag Potkonjak*

*University of California, Los Angeles
Los Angeles, CA 90095, USA

†California State University, Northridge
Northridge, CA 91330, USA

shengwei@cs.ucla.edu, jwendt@cs.ucla.edu, ani@csun.edu, miodrag@cs.ucla.edu

ABSTRACT

This paper investigates and addresses the vulnerabilities of existing physical unclonable functions (PUFs). We first develop a PUF reverse engineering approach by conducting gate-level characterization (GLC). Based on the gate-level delay properties, we emulate the target PUF by designing a functionally equivalent PUF replication. Furthermore, in order to prevent such an attack, we develop a new sequential PUF architecture that is resilient to side channel-based reverse engineering. We obtain accurate results in emulating the timing behavior of the existing arbiter-based PUFs. Also, the randomness obtained from the sequential PUFs is significantly higher compared to the existing PUFs.

1. INTRODUCTION

A silicon physical unclonable function (PUF) is a multi-input multi-output device whose input-output mapping is difficult to predict and reverse engineer and thus impossible to clone. Recently, research on PUF design has drawn a great deal of attention, because the PUF can serve as a new type of security platform that is resilient against various attacks, such as side channel, physical, and software attacks. Furthermore, PUFs are orders of magnitude more efficient than classical cryptography techniques in terms of energy consumption and performance.

Considering the importance of PUFs in a variety of hardware security applications, several reverse engineering (RE) attempts have been made to either challenge or evaluate the existing PUFs [8][15][22][23]. For example, Majzoobi et al. [8] proposed a PUF reverse engineering approach using linear algebra and numerical analysis. Ruhrmair et al. [15] employs statistical and machine learning techniques to reverse engineer PUFs. However, these reverse engineering efforts only targeted a limited number of specific PUFs. Also, they did not address the issues of fast emulation and physical realization of the reverse engineered PUFs. Instead, the prediction was conducted at the software and simulation level, which significantly impacts its effectiveness.

We first develop a side-channel based reverse engineering method to challenge the security of the existing PUF designs. We employ gate-level characterization (GLC) techniques that require a limited number of global leakage power measurements and extract the gate-level physical properties, such as threshold voltage and effective channel length. Based on the physical properties, we are able to recover the resulting delay properties of the PUF components that are crucial to the behavior of the PUF. Then, we show that the existing PUF designs, which leverage the unpredictability of the delta delay between two orthogonal delay paths, can be effectively reverse engineered and reproduced using emulations, considering the low measurement and characterization errors in our GLC process.

Furthermore, in order to prevent such a side channel and emulation-based reverse engineering attack, we design a new generation of PUFs that exploits multiple sequential executions of the combinational PUF logic. We demonstrate that the sequential PUF design has a extremely low probability to be reverse engineered compared to the existing PUF designs, because of the significantly increased randomness and unpredictability. To summarize, we have the following three main technical contributions:

- We develop a side-channel based reverse engineering attack against existing arbiter-based PUFs, in which we recover the delay properties of the PUF components via physical gate-level characterization;
- We demonstrate an accurate PUF emulation approach based on GLC that reproduces the exact behavior of a wide class of existing PUFs; and
- We design a RE-resilient PUF that employs sequential executions of the combinational PUF logic.

2. RELATED WORK

Silicon physical unclonable functions emerged in 2002 as a powerful hardware security primitive for ultra low power embedded systems [5]. However, PUFs by nature are secret keys and require storage of challenge-response pairs. In 2009, Beckmann et al. introduced the public PUF (PPUF) to solve the protocol limitations of PUFs [2]. Here, physical-level properties that determine the input-output mapping (e.g. gate delays) are published and serve as a public key. Although PPUFs solved various security and protocol limitations of traditional PUFs, their operation requires that one involved party (e.g. the authenticator) has significant computation resources and time.

DAC'14, June 01 - 05 2014, San Francisco, CA, USA
Copyright 2014 ACM 978-1-4503-2730-5/14/06 ...$15.00.
http://dx.doi.org/10.1145/2593069.2593204

Most silicon PUFs and PPUFs derived their security properties from process variation (PV) [2][13][14]. UCLA researchers leveraged both PV and device aging to create the matched PPUF (mPPUF), where two unique mPPUF instances are configured to realize identical input-output mappings through a combination of coordinated device aging and gate disabling [11][12]. The mPPUF enables essentially single-cycle public key communication protocols, but it is limited in the sense that one must essentially possess a separate mPPUF instance for every other party to which one wishes to communicate.

3. PRELIMINARIES

3.1 Process Variation

Process variation (PV) in IC manufacturing is the deviation of IC parameter values from nominal specifications[1][3]. It causes major variations in gate-level physical properties, such as L_{eff} and V_{th}. PV is an unavoidable technological phenomenon of all deep submicron and nano IC realization technologies. The main PV ramification is that each device (e.g. gate, transistor, and interconnect) of the same design has different manifestational characteristics (e.g. delay or static power) on different ICs. These device level characteristics have profound impact on the overall IC characteristics.

3.2 Power and Delay Models

We consider leakage power, switching power, and delay as manifestational properties of an IC. There are two sources for power dissipation in an IC. One is from gate switching, in which the ICs dissipate power by charging the load capacitances of wires and gates. The other source is leakage power, where even if the gates do not switch, they dissipate power due to the leakage current. Equation (1) is the gate-level leakage power model [10], where W is gate width, L is gate length, V_{th} is threshold voltage, V_{dd} is supply voltage, and T is the temperature. The rest of the parameters are considered as constants, which are discussed in details in [10]. We observe from Equation (1) that the leakage power is dependent on L and V_{th} in a non-linear manner.

$$P_{leakage} = 2 \cdot n \cdot \mu \cdot C_{ox} \cdot \frac{W}{L} \cdot (\frac{kT}{q})^2 \cdot D \cdot V_{dd} \cdot e^{\frac{\sigma \cdot V_{dd} - V_{th}}{n \cdot (kT/q)}} \quad (1)$$

The gate-level switching power model [16] is described by Equation (2), where the switching power is dependent on switching probability α, gate width W, gate length L, and supply voltage V_{dd}.

$$P_{switching} = \alpha \cdot C_{ox} \cdot W \cdot L \cdot V_{dd}^2 \quad (2)$$

Equation (3) shows the gate-level delay model [10] that is dependent on the parameters similar with leakage power. Also similarly, gate delay depends on the L and V_{th} in a non-linear manner. Load capacitance C_L is defined in Equation (4), where γ is the logical effort of the gate, and W_{fanout} is the sum of the widths of the load gates.

$$Delay = \frac{k_{tp} \cdot C_L \cdot V_{dd}}{2 \cdot n \cdot \mu \cdot C_{ox} \cdot \frac{W}{L} \cdot (\frac{kT}{q})^2} \cdot \frac{k_{fit}}{(ln(e^{\frac{(1+\sigma)V_{dd} - V_{th}}{2 \cdot n \cdot (kT/q)}} + 1))^2} \quad (3)$$

$$C_L = C_{ox} \cdot L \cdot (\gamma \cdot W + W_{fanout}) \quad (4)$$

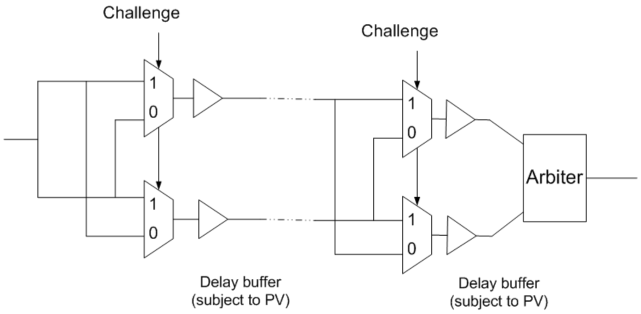

Figure 1: Arbiter-based PUF design [5][7][9][17].

3.3 Arbiter-based PUF

Figure 1 shows the existing popular arbiter-based PUF architecture [5][7][17], which serves as the baseline of this paper. In this PUF design, each challenge bit selects one of the two multiplexers and thus one of the two delay buffers that generate different propagation delay values due to process variations. After going through n stages of the delay buffers, the accumulated delay is expected to have a significant level of randomness that is difficult to predict by an attacker. Based on this basic arbiter-based architecture, many researchers [2][5][11] have focused on how to further increase the randomness and reduce the probability of emulating and predicting the response outputs from the given challenge bits.

4. REVERSE ENGINEERING USING GATE-LEVEL CHARACTERIZATION

The goal in PUF reverse engineering is to accurately estimate the gate-level property of the IC, such as delay, which is used by the PUF to generate random outputs. Based on the estimations, one can intentionally emulate the target PUF using the replicated copy. In this section, we discuss our approach of conducting PUF reverse engineering using gate-level delay properties.

4.1 Overall Flow

It is important to note that the gate-level delay properties of a manufactured chip cannot be directly measured due to the physical constraints of the IC and high costs of conducting those measurements. Therefore, we must leverage only the global measurements of IC properties, such as leakage power, switching power, and/or delay to recover the gate-level delay properties.

Our approach of gate-level characterization is to measure global leakage power consumption of the IC and identify the physical properties of each gate, such as effective channel length and threshold voltage, which are subject to process variation. The characterized gate-level properties can serve as a foundation of many hardware-based security applications. In particular, from the reverse engineer's perspective, if one can obtain the accurate physical properties of a PUF, it would be possible for an attacker to duplicate the physical copy or, at least, emulate the exact behavior of the designed PUFs. Consequently, the integrity and security of the PUFs may be compromised.

We develop a gate-level characterization model that can be leveraged by an attacker and serves as the foundation

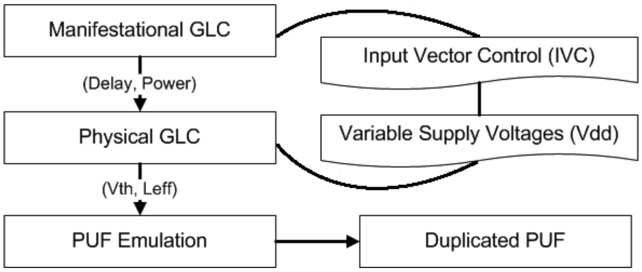

Figure 2: Overall flow of the GLC-based reverse engineering approach.

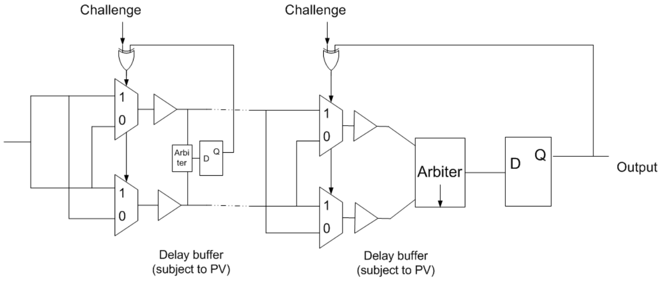

Figure 3: Architecture of sequential PUFs.

for reverse engineering the existing PUFs. In particular, we characterize the two key gate-level properties, namely threshold voltage and effective channel length, which have fundamental impacts on the manefistational IC properties and are deviated from the nominal specifications after manufacturing due to process variation. Figure 2 shows the overall flow of our gate-level characterization approach. We first characterize the gate-level manifestational properties, such as delay and power, by formulating a system of linear equations. Then, we obtain gate-level physical properties (e.g., threshold voltage and effective channel length) by using the GLC results and by manipulating the inputs and voltages. Finally, we emulate the PUF behavior based on the GLC results and create the functionally equivalent attacker PUF to trigger a variety of security attacks.

4.2 Means to Creating Independent Equations

One of the challenges in characterizing gate-level properties via only global IC measurements is to create multiple independent equations concerning the overall leakage power consumption and that of individual gates [19]. We leverage two types of techniques to create these equations. The first technique is based on the fact that the leakage power of a logic gate highly depends on its input signals [21]. Therefore, we can create varying leakage power profiles by varying the input vectors of the IC. This is the fastest and most efficient way of obtaining variations and creating various equations. However, as pointed out by [19], input vector control is not sufficient in creating systems of equations that are solvable, due to the fact that collinearity of the coefficients and insufficient controllability both exist in most of the IC designs. In order to address this issue, we employ three additional approaches that result in non-correlated variations in the gate-level leakage power consumption, namely variable supply voltage, thermal conditioning, and device aging.

- *Variable supply voltages.* According to the power model (described in details in Section 3.2), the threshold voltage being applied during the IC operation plays an important role in the leakage power consumption of a logic gate. In particular, Keshavarzi et al. [6] has evaluated the non-linear impact of threshold voltage on the leakage current. Therefore, in order to create multiple linearly independent equations, we can vary the supply voltage of the circuit and keep repeating the measurements.

- *Thermal conditioning.* We use a thermal conditioning technique, which is based on the following fact: the leakage power grows exponentially as the temperature increases. This observation is derived from Equation (1), where the leakage power is dependent on temperature T in a nonlinear manner. The relation between leakage power and temperature provides good intuition of how to obtain linear equations with different sets of coefficients in our linear program formulation.

- *Intentional device aging.* IC aging is the process of stressing the transistors in the circuit, where the consequence is the increase of threshold voltage over the stress time [4] [18]. Aging has been long considered as a detrimental effect to the IC lifetime. However, we can leverage the increase of threshold voltage caused by aging to vary the leakage current of the entire IC in a non-linear manner.

Based on the global power measurements and variations obtained from the above techniques, we create linear and non-linear programs to solve the gate-level manifestational and physical properties, respectively. The details of the two GLC processes can be found in [19] and [20].

4.3 PUF emulation

After obtaining the gate-level properties via the GLC process, we continue with emulating the original PUFs to predict the response bits under the given challenges. In particular, we create an attacker PUF that exactly emulates the behavior of the original PUF by employing the characterized PUF's properties. Since the netlist of the original PUF is known, we can accurately simulate the PUF behavior by plugging in the gate-level values into a simulator, calculate the actual propagation path determined by the challenge, and obtain the corresponding output signals.

5. RE-RESILIENT PUFS DESIGN

As a defense, we develop a new PUF architecture that is resilient to the aforementioned GLC-based reverse engineering. In this section, we discuss in details our design objectives and the new RE-resilient PUF architecture.

5.1 Design Objectives

The goal of the RE-resistant PUF design is to prevent the reverse engineering approach from extracting the physical implementation details of the PUF, replicating it with an equivalent faked hardware implementation, and later emulating the exact responses with the faked copy. We achieve this goal using the following two strategic steps:

- *Complicating the RE characterization.* We design the PUF architecture in such a way that it creates the

Table 1: Accuracy of Gate-level Characterizations on Existing PUFs.

PUF Design	Approach	Number of Gates	GLC accuracy (V_{th})	GLC accuracy (L_{eff})
Gassend et al. (2002) [5]	Switch-based ring oscillator	24	1.05%	0.48%
Majzoobi et al. (2009) [9]	Light-weight arbiter	256	1.11%	0.50%
Suh et al. (2007) [17]	Inverter ring oscillator	120	1.14%	0.51%
Lee et al. (2004) [7]	Feed-forward arbiter	196	1.25%	0.65%

most challenging cases for the existing side channel-based IC characterization approaches. With these designs, we ensure that the difficulty level of obtaining the gate-level properties out of the PUF is significant. Also, even if the PUF is still reverse engineered, it is difficult to achieve high levels of accuracy. Consequently, it is computationally challenging to accurately emulate the PUF due to the inaccurate characterizations, as any inaccuracies in the intermediate stages are highly transitive and would indirectly impact the eventual response outputs.

- *Complicating the RE emulation.* The effectiveness of PUF reverse engineering is highly dependent on the RE emulation step, where the attacker counterfeits the target PUF by either replicating its physical architecture or by implementing an equivalent design that has exactly the same behavior as the original PUF. While replicating the PUF physically might be difficult, we have shown in the previous sections that emulating the PUF at the functional level is doable assuming the attacker has the ability of extracting the gate-level properties of the target PUF in the GLC process. This poses significant challenges in achieving a RE-resilient PUF design.

5.2 Sequential PUF Architecture

Our approach to complicating the RE emulation is to design a new PUF architecture that employs time-based sequential events, namely a sequential PUF. Figure 3 shows our sequential PUF architecture. The PUF execution flow involves multiple sequential runs of the combinational arbiter-based PUF logic. Between different runs, we feed a set of the intermediate outputs (e.g., from randomly selected intermediate stages) into a XOR logic, which combines selected signals with the initial challenge, to form the challenge of the next run. The eventual response outputs of the sequential PUF are those obtained after K sequential runs, where K is a configurable parameter based on the timing and randomness requirements of a specific use case.

6. EVALUATION RESULTS

In this section, we evaluate our PUFs reverse engineering and anti-RE PUFs design. In particular, we implement the RE techniques on a set of popular PUF architectures, including [5][7][17][9]. Also, we evaluate our RE-resistant PUFs design by calculating the hamming distance between the response bits.

6.1 GLC on Existing PUFs

We evaluate the GLC-based reverse engineering approach on a variety of popular PUFs, as shown in Table 1. For each PUF, we conduct physical GLC, in terms of threshold voltage and effective channel length, and compare the characterized results against the actual property values. We use the relative characterization errors, i.e., the percentage that the characterized value is apart from the actual value, to represent the accuracy of the reverse engineering. The characterization errors serve as indicators of the accuracy that an attacker could achieve to conduct emulation and reproduce the behavior of the target PUF. From Table 1 we observe that the resulting characterization errors are below or around 1% for a set of existing PUFs.

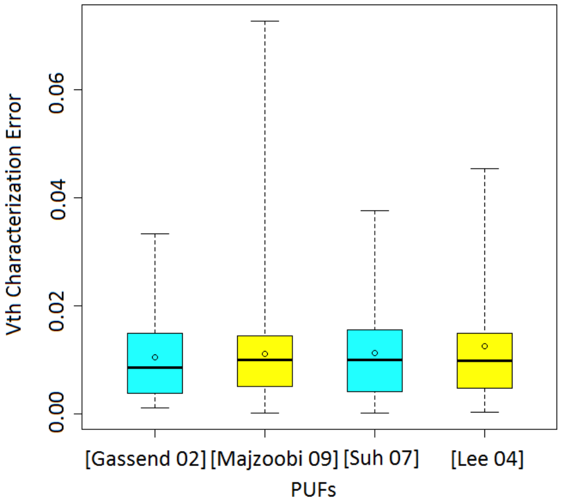

Figure 4: Distribution of threshold voltage characterization errors.

Furthermore, in Figure 4 and Figure 5 we plot the distributions of the GLC errors for threshold voltage and effective channel length, respectively. We observe that the average errors of GLC fall within the range below 2%, which ensures the accuracy of the reverse engineering approach.

6.2 PUFs Emulation

We evaluate the accuracy of our PUF emulation approach by comparing the outputs of the attacker PUF and the original PUF. In particular, we first construct a certain number (e.g., 50) of original PUFs with a Gaussian distribution of V_{th} and apply a certain number (e.g., 50) of challenges to those PUFs and record their outputs as the baseline of correct responses. Then, we construct a certain number (e.g., 100) of attacker PUFs per original PUF considering the characterization errors from GLC. For each original PUF

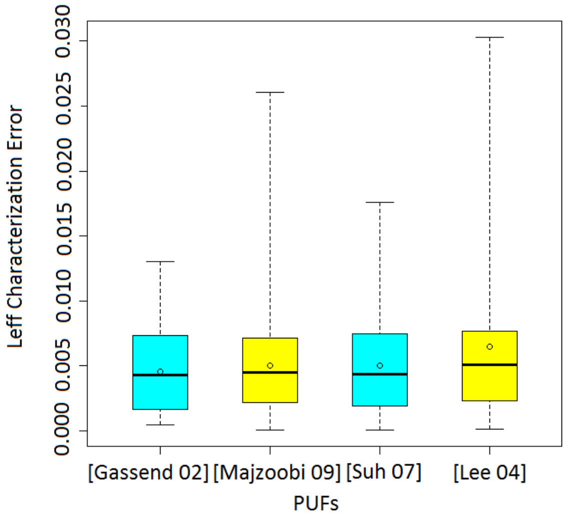

Figure 5: Distribution of effective channel length characterization errors.

and challenge we emulate the 100 attacker PUFs and compute the incorrect response rate compared to the correct responses.

Figure 6 shows the distribution of incorrect responses of the attacker PUFs using our PUF emulation approach. We simulate 100 attacker PUFs that are created using GLC errors (i.e., V_{th} error rate 1.05% and L_{eff} error rate 0.48%) per 50 PUF instances and 50 challenges. Also, the PUF sizes vary between 5 configurations of $M \times N$, where M is the number of stages (i.e., challenge bits), and N is the number of rows (i.e., response bits) in the PUF structure. We observe that more than 90% of the attacker PUFs using our emulation approach have zero error rates for all the single-output PUFs (i.e., $1 \times 64, 1 \times 128$ and 1×256). With the increase of N (i.e., the number of response bits), the error rates increase, but we still have more than 25% of the attacker PUFs that are error free. Considering that even a smaller percentage of correct attacker PUFs will cause huge damage to the PUF security, we conclude that the proposed GLC-based PUF emulation approach is effective.

Similarly, we evaluate the impact of process variation on the PUF emulation accuracy, as shown in Figure 7. We simulate the response error rates of the attacker PUFs (1×256) with various variances for the V_{th} Gaussian distribution. The results indicate that the accuracy of our emulation approach is resilient to the process variation, as there are over 90% of the attacker PUFs generating the correct outputs in all the tested variances.

6.3 RE-Resilient Sequential PUFs

Figure 8 and Figure 9 show the distributions of the output hamming distances in the existing PUF [5] and in our sequential PUF design, respectively. In our experiments, we evaluated 25 pairs of challenges, with each pair differing by only 1 bit of the inputs. The hamming distance between the resulting outputs serve as an indicator of how effective the PUF is. For example, the larger the hamming distance is, the less correlation the output has with the challenge bits,

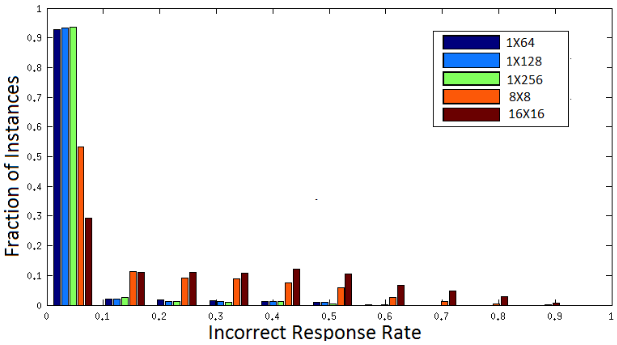

Figure 6: Distribution of error rates in our PUF emulation with varying PUF sizes.

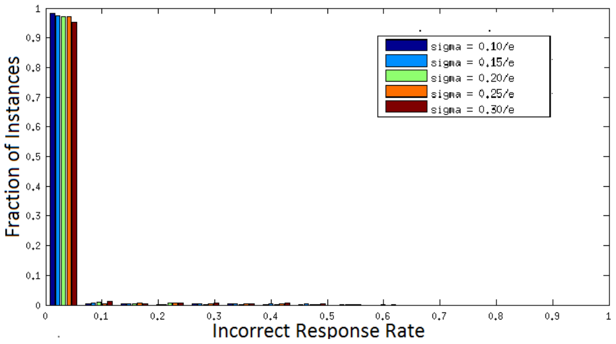

Figure 7: Distribution of error rates in our PUF emulation with varying process variations.

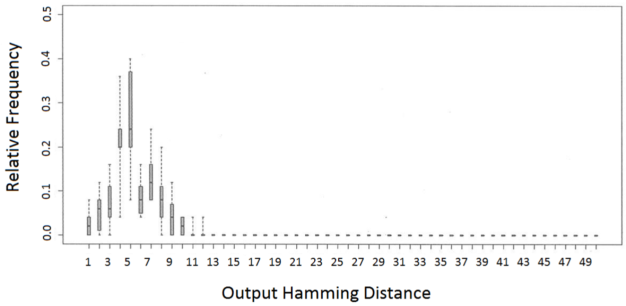

Figure 8: The distribution of output hamming distance (without sequential loop).

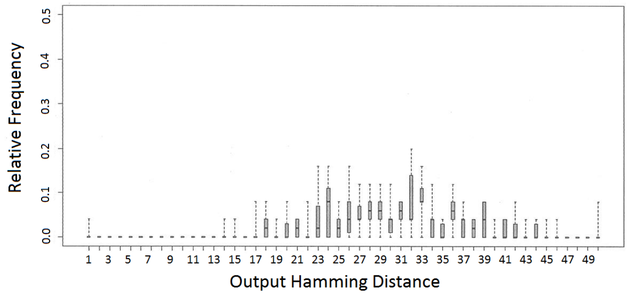

Figure 9: The distribution of output hamming distance (with sequential loop).

which means it is more difficult to be reverse engineered. From Figure 8 and Figure 9, we observe that the hamming distances in our loop-based sequential PUFs are significantly larger than those of the existing PUFs. This proves the randomness and thus the effectiveness of our sequential PUF, in the sense that it can generate significantly random outputs with only 1 bit variation in the challenge bits. This prevents attackers from engineering the PUF by varying the challenge bit by bit and observing the output patterns.

7. CONCLUSION

We investigated the security vulnerabilities of the existing arbiter-based PUFs by conducting side channel-based reverse engineering and emulation. With accurate reverse engineering results, we argue that the existing arbiter-based PUF designs have security vulnerabilities that must be addressed. To solve this issue, we developed a sequential PUF architecture that is resilient to side channel-based reverse engineering. Our evaluation results show that the proposed reverse engineering attack has an over 90% success rate. Also, our sequential PUF mechanism significantly reduces the predictabilities of the PUFs and thus increases the level of difficulty for reverse engineering attacks.

8. ACKNOWLEDGEMENTS

This work was supported in part by the NSF under Award CNS-0958369, Award CNS-1059435, and Award CCF-0926127, and in part by the Air Force Award FA8750-12-2-0014.

9. REFERENCES

[1] A. Asenov. Random dopant induced threshold voltage lowering and fluctuations in sub-0.1 μm MOSFET's: A 3-D "atomistic" simulation study. *IEEE Transactions on Electron Devices*, 45(12):2505–2513, 1998.

[2] N. Beckmann and M. Potkonjak. Hardware-based public-key cryptography with public physically unclonable functions. In *Information Hiding (IH)*, pages 206–220, 2009.

[3] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De. Parameter variations and impact on circuits and microarchitecture. In *DAC*, pages 338–342, 2003.

[4] S. Chakravarthi, A. Krishnan, V. Reddy, C. Machala, and S. Krishnan. A comprehensive framework for predictive modeling of negative bias temperature instability. In *IRPS*, pages 273–282, 2004.

[5] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas. Silicon physical random functions. In *CCS*, pages 148–160, 2002.

[6] A. Keshavarzi, K. Roy, and C. Hawkins. Intrinsic Leakage in Low-Power Deep Submicron CMOS ICs. In *ITC*, pages 146–155, 1997.

[7] J. Lee, D. Lim, B. Gassend, G. Suh, M. van Dijk, and S. Devadas. A technique to build a secret key in integrated circuits for identification and authentication applications. In *Symposium on VLSI Circuits*, pages 176–179, 2004.

[8] M. Majzoobi, F Koushanfar, and M. Potkonjak. Testing techniques for hardware security. In *ITC*, pages 1–10, 2008.

[9] M. Majzoobi, F. Koushanfar, and M. Potkonjak. Techniques for design and implementation of secure reconfigurable PUFs. *ACM Transactions on Reconfigurable Technology and Systems*, Vol. 2, No. 1, Article 5, 5:1–5:33, 2009.

[10] D. Markovic, C. Wang, L. Alarcon, T. Liu, and J. Rabaey. Ultralow-power design in near-threshold region. *Proceedings of the IEEE*, 98(2):237–252, 2010.

[11] S. Meguerdichian and M. Potkonjak. Matched public PUF: Ultra low energy security platform. In *ISLPED*, pages 45–50, 2011.

[12] S. Meguerdichian and M. Potkonjak. Using standardized quantization for multi-party PPUF matching: Foundations and applications. In *ICCAD*, page 577–584, 2012.

[13] J. B. Wendt and M. Potkonjak, Nanotechnology-based trusted remote sensing. in *IEEE Sensors*, pages 1213–1216, 2011.

[14] J. B. Wendt and M. Potkonjak, The bidirectional polyomino partitioned PPUF as a hardware security primitive. in *GlobalSIP*, pages 257-260, 2013.

[15] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber. Modeling attacks on physical unclonable functions. In *CCS*, pages 237–249, 2010.

[16] A. Srivastava, D. Sylvester, and D. Blaauw. *Statistical Analysis and Optimization for VLSI: Timing and Power*. Springer, 1999.

[17] G. Suh and S. Devadas. Physical unclonable functions for device authentication and secret key generation. In *DAC*, pages 9–14, 2007.

[18] W. Wang, S. Yang, S. Bhardwaj, R. Vattikonda, S. Vrudhula, F. Liu, and Y. Cao. The impact of NBTI on the performance of combinational and sequential circuits. In *DAC*, pages 364–369, 2007.

[19] S. Wei, S. Meguerdichian, and M. Potkonjak. Gate-level characterization: Foundations and hardware security applications. In *DAC*, pages 222–227, 2010.

[20] S. Wei, M. Potkonjak. Integrated circuit security techniques using variable supply voltage. In *DAC*, pages 248–253, 2011.

[21] L. Yuan and G. Qu. A combined gate replacement and input vector control approach for leakage current reduction. *IEEE Transactions on VLSI Systems*, 14(2):173–182, 2006.

[22] U. Rührmair, X. Xu, J. Sölter, A. Mahmoud, F. Koushanfar, and W. Burleson. Power and timing side channels for PUFs and their efficient exploitation. *IACR Cryptology ePrint Archive*, Report 2013:851, 2013.

[23] A. Mahmoud, U. Rührmair, M. Majzoobi, and F. Koushanfar. Combined modeling and side channel attacks on strong PUFs. *IACR Cryptology ePrint Archive*, 2013:632, 2013.