Matched Public PUF: Ultra Low Energy Security Platform

Saro Meguerdichian and Miodrag Potkonjak
Computer Science Department
University of California, Los Angeles
{saro, miodrag}@cs.ucla.edu

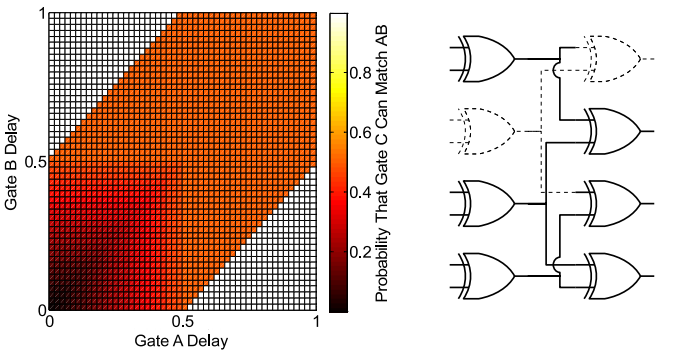
***Abstract*—Hardware-based physically unclonable functions (PUFs) leverage intrinsic process variation of modern integrated circuits to provide interesting security solutions but either induce high storage requirements or require significant resources of at least one involved party. We use device aging to realize two identical unclonable modules that cannot be matched with any third such module. Each device enables rapid, low-energy computation of ultra-complex functions that are too complex for simulation in any reasonable time. The approach induces negligible area and energy costs and enables a majority of security protocols to be completed in a single or a few clock cycles.**

***Index Terms*—hardware security; public key cryptography; device aging; PUF; PPUF**

I. Introduction

Our strategic objective is to introduce concepts and a hardware platform that enable ultra low power security protocols. Security plays an essential role in numerous applications. Wireless system security imposes additional system requirements and greatly amplifies some of the existing constraints. For example, resiliency against physical and side channel attacks and low energy consumption are among the mandatory design and operation metrics. The scope of security recently has been tremendously increasing from initial concerns about privacy of communication and stored data and authentication to digital rights management, trust, privacy, and physical and social security issues [1] [2] [3] [4]. Traditional mathematical cryptography has demonstrated both its algorithmic soundness and industrial practicality in many domains. However, for large classes of wireless applications, including sensor networks and personal mobile communication, the traditional security techniques and system are increasingly often not adequate.

Process variation (PV) is an unavoidable side product of modern and pending silicon implementation technologies. As a ramification of PV, each transistor, gate, and wire on each integrated circuit that realizes a particular design has unique physical (e.g. channel length) and manifestational (e.g. leakage energy and delay) properties. A physically unclonable function (PUF) is a deterministic multiple-input multiple-output system that is hard to reverse engineer and simulate. Silicon PUFs are currently by far the most popular and most effective hardware-based security systems that are intrinsically resilient to physical and side channel attacks. Major PUF limitations include the use of secret key cryptography, large storage space requirements, and the ability to realize only a very limited set of security protocols. Recently developed public-key PUF schemes such as public physically unclonable functions (PPUFs) and SIMPL remove all these limitations but require that at least one participating party is capable of conducting complex simulations. Therefore, low power and low latency applications cannot be realized using these techniques.



(a) (b)

Fig. 1: Motivational example: (a) probability that a third, malicious gate C can match two honestly matched gates A and B; (b) small mPPUF with 6 matched and 2 disabled gates.

Our technical goal is to demonstrate a new class of PUFs that preserves all advantages of PPUFs but requires each party in a wide class of security protocols to conduct only a single cycle computation. Therefore, in a sense, our goal is to create an ultimately low energy and low latency security approach that enables the device to operate in hostile environments. The key idea is to use device aging (e.g. NBTI-induced transistor slowdown or wire electromigration) to create two completely identical PPUFs in such a way that the probability that a third PPUF can have the same characteristics are negligible.

Consider, for example, two PPUF owners Alice and Bob who want to match their PPUFs. Assume a very simple PV model for delay where individual gate delays follow a uniform random distribution between 0 and 1, and an aging model where maximal aging of a gate increases its delay by 0.5. Without loss of generality, consider the following cases when aging a gate A with delay D_A on Alice's PPUF to match its corresponding gate B with delay D_B on Bob's PPUF:

1) $D_A = D_B$. This is the trivial case. Gates A and B have equal delay, so they are already matched.

2) $D_B - 0.5 \leq D_A < D_B$. Here, gate A is faster than gate B, but not by more than 0.5. To match the gates, Alice ages gate A by $\Delta D_A = D_B - D_A$ such that $D_A + \Delta D_A = D_B$.
3) $D_A < D_B - 0.5$. In this case, gate A is faster than gate B by more than 0.5. Hence, gates A and B cannot be matched and must be disabled.

To match their PPUFs, Alice and Bob both follow the same aging and disabling procedure outlined above. Therefore, the only cases for which gate A and gate B cannot be matched are when $D_A < D_B - 0.5$ or $D_B < D_A - 0.5$. It is easy to see that the probability of either of these events occurring is $1/4$, shown as the white regions in Figure 1a. Therefore, Alice and Bob are able to match 75% of their gates, on average.

Now, assume that a third, malicious PPUF owner, Chuck, wants to match his PPUF to the same configuration as that matched by Alice and Bob. The following two cases are possible, for a gate C with delay D_C on Chuck's PPUF:

1) *Gates A and B were not able to match and were disabled.* In this case, Chuck must simply disable gate C to match the configuration.
2) *Gates A and B were matched.* Here, Chuck will only be able to age gate C to match if it is faster than the slowest gate between A and B, but not by more than 0.5 (i.e. $\max(D_A, D_B) - 0.5 \leq D_C < \max(D_A, D_B)$).

Figure 1a shows the probability that Chuck is able to match an unwelcome gate C with two honestly matched gates A and B for various delays D_A and D_B. This occurs with probability $7/12$; in other words, Chuck is able to match only 58.33% of the configuration matched honestly by Alice and Bob.

After matching their PPUFs (e.g. Figure 1b), Alice and Bob now have PPUFs that realize the same complex function. In other words, both PPUFs (and no other PPUFs) will produce exactly the same unique response to any challenge in a single cycle. Therefore, Alice can issue Bob a challenge and verify his response by executing it on her own PPUF, enabling a myriad of low-energy cryptographic protocols that require neither high storage nor simulation.

To the best of our knowledge, the matched PPUF (mPPUF) is the first ultra low power cryptographic primitive and implementation that requires that all participating parties require only a single cycle energy for security protocols such as authentication, message authentication, and public key communication. Also, it is the first scheme that requires only self-trust: each party has control over announcing its own public key using preliminary aging. Through public key elimination of storage and elimination of simulation, the mPPUF combines the best properties of both PUFs (single cycle operation and low energy) and PPUFs (public key security and no storage requirements).

II. Related Work

In this section we briefly summarize the most directly related literature in process variation (PV), device aging, and PUFs.

A. Process Variation

It has been widely recognized that modern ICs are unique due to factors such as line edge roughness, polysilicon granularity, and random discrete dopants [5]. Numerous transistor- and gate-level characterization (GLC) techniques, the enabling technologies for PPUFs, have been proposed, including: (i) direct measurements approaches [6]; (ii) schemes that employ FPGA reconfiguration [7]; (iii) approaches that create and observe special IC structures and specialized circuitry [8]; and (iv) non-destructive techniques that construct global measurements and deduce scaling factors of each gate by solving a system of equations [9] [10] [11] [12]. We use transistor and spatial correlation from [5] and [13], respectively.

B. Device Aging

Negative bias temperature instability (NBTI) and hot carrier injection (HCI) are examples of intrinsic phenomena of deep submicron silicon technologies that have detrimental impacts on reliability and speed of operation [14]. Recently, they have attracted a great deal of attention mainly due to reliability issues. NBTI effects are in particular pronounced for PMOS transistors. Its impact continues to increase with each technology node [5].

C. PUFs and PPUFs

A great impetus for hardware based security was provided by a MIT paper that introduced the first PUF using a mesoscopic optical system [15]. Soon, again at MIT, Devadas and his group proposed the silicon PUF concept and demonstrated its ASIC realization [16]. More recently, Beckmann et al. proposed the first public PUF scheme [17]. A conceptually similar but technically drastically different approach was proposed at the University of Munich [18].

Traditional PUF and PPUF schemes leverage PV [19] [20] [21]. Recently proposed device aging-based PUFs and PPUFs leverage both PV and device aging [22] [23]. The mPPUF ensures complete trust self-sufficiency, one cycle ultra low power operation, resiliency against physical and side channel attacks, and high security flexibility.

III. Preliminaries

A. Gate Delay, Power, and Process Variation

We use the gate-level delay and power models from Markovic et al. [24]. The delay model is reproduced in Equation 1, where k_{tp} is the delay-fitting parameter, C_L is load capacitance, V_{dd} is supply voltage, n is subthreshold slope, μ is mobility, C_{ox} is oxide capacitance, W is gate width, L is effective channel length, $\phi_t = kT/q$ is thermal voltage, k_{fit} is a model-fitting parameter, σ is the drain induced barrier lowering (DIBL) factor, and V_{th} is threshold voltage.

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

There are two parameters that are directly impacted by PV: effective channel length (L) and threshold voltage (V_{th})

[25]. For V_{th}, we adopt the Gaussian distribution proposed by Asenov et al. [26]. Note that gates are not correlated in terms of V_{th}. For L, however, we follow the quad-tree model proposed by Cline et al. [13], which considers the spatial correlations among gates.

B. Device Aging

We use the aging model proposed by Chakravarthi et al. [27] and shown in Equation 2 for the effect of device aging due to NBTI on V_{th} shift, where A and β are constants, V_G is the applied gate voltage, E_α is the measured activation energy of the NBTI process, T is the temperature, and t is time.

$$\Delta V_{th} = A \cdot e^{\beta V_G} \cdot e^{-E_\alpha/kT} \cdot t^{0.25} \tag{2}$$

For mPPUF matching, we use static (DC) aging, which can be reversed by removing the applied stress [14]. Note that the model follows a fractional power law; in other words, a relatively large amount of aging happens in a relatively short amount of time, when the input vectors are first applied.

IV. Desiderata

We begin the discussion of the mPPUF by identifying the architectural, operational, and security desiderata. Architectural desiderata include (i) low energy, delay, and area costs; (ii) stability against temperature and voltage variations; and (iii) suitability for inexpensive, in-field, and accurate characterization. Operational desiderata include (iv) fast and low-energy aging for mPPUF matching; (v) the ability to match arbitrary mPPUF instances; and (vi) indefinite reconfigurability. Finally, security desiderata include (vii) resiliency against security attacks; (viii) intractably large simulation time; and (ix) low probability of coincidence.

V. Architecture

The mPPUF architecture shown in Figure 2 has width w and consists of s stages. Each stage is comprised of b k-input booster cells and r k-input represser cells, the details of which we discuss in Section V-A. The design has height $h = s{\cdot}(b{+}r)$, where the w cells in each sequential level take as inputs the outputs of the k cells in the previous level. A final level of w 1-input terminator cells are added to enhance stability.

Inputs from w flip-flops race against the clock through the various input-output paths in the circuit to w arbiters. An arbiter has 2 inputs, x and y; it outputs 0 if input x transitions from $0 \to 1$ before input y and 1 otherwise. In other words, each arbiter outputs 1 if and only if the first $0 \to 1$ transition of its input (from the final stage) occurs before the $0 \to 1$ clock transition.

Output unpredictability, difficulty of simulation, and matching ability are among our most crucial goals. Therefore, the interconnection network between consecutive levels must provide a high degree of signal mixing. In other words, in an ideal interconnection network: (i) each input drives the same number of gates (k); (ii) no two gates share many inputs; and (iii) after h levels, each output depends on all inputs.

A. Cells

As discussed in the previous subsection, each stage is composed of b levels of booster cells followed by r levels of represser cells. A booster's role is to increase the switching frequency of its output signal in relation to its input signals by a factor of B. The role of a represser cell is the opposite, to reduce the switching frequency by a factor of R. Therefore, intuitively, a cell with k inputs is a booster if, on average, more than $1/k$ of the switches of each input causes it to switch; otherwise, the cell is a represser.

The combination of booster and represser cells is what creates the enormous simulation complexity of the mPPUF. Booster cells increase the number of output transitions exponentially with the number of levels. Specifically, after b levels of boosters with boost factor B, the output switches by a factor b^B more than the input. Represser cells complement booster cells by unpredictably repressing frontier signal transitions that would otherwise lock the arbiters.

1) Boosters: We show that the optimal k-input booster cell is a k-input XOR gate. Consider, for example, a 2-input XOR gate. It is clear that for this gate, any time either of the 2 inputs transitions from $0 \to 1$ or $1 \to 0$, the output will transition as well. Therefore, a 2-input XOR gate has boosting factor $B = 2$. It follows that a k-input XOR gate is a booster cell with boosting factor $B = k$. Note that without combinational loops, we cannot have a boosting factor $B > k$.

2) Repressers: The ultimate represser cell is one which never switches (i.e. $R = \infty$); however, such a cell would terminate the propagation of all signals and thus render the device useless. Therefore, a good represser is one which represses switching to enough of a degree that a high but unpredictable number of frontier signals are repressed.

Consider, for example, a 4-input NAND gate, which would seem at first approximation to have repression factor $R = 1/8$, since out of a possible 64 input transitions, the output will switch for only 8: from 1111 to any input (4) or from any input to 1111 (4). Indeed, if the inputs to the NAND gate are random, then the gate does repress with factor $R = 1/8$.

Now consider the case where we have two consecutive levels of 4-input NAND gates, with random inputs to the first level. The first level indeed represses with factor $R = 1/8$. However, these gates drive most of the signals to be most often 1 (roughly 93.75% of the time). As a result, the inputs to any of the represser cells in the second level are very likely to be in one of the five transition cases (four 1's or three 1's). Therefore, the second-level represser cells actually act as booster cells with boost factor roughly $B = 2$, obtained from simulation.

In light of this observation, we use different represser cells at consecutive levels. Specifically, we alternate between the four represser cells with Karnaugh maps shown in Table I. These repressers still drive the output to 1, but create 0's for different combinations of inputs in a balanced way around 1111. Simulation results show that the represser cells maintain an average repression factor of roughly $R = 1/8$ when

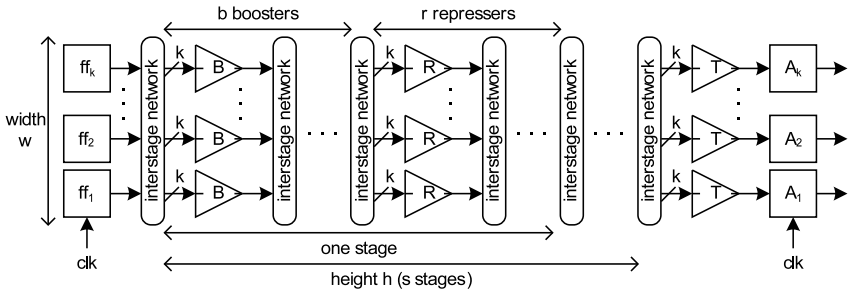


Fig. 2: mPPUF architecture of width w and height h, consisting of s stages of b boosters (B) and r repressers (R), with a final level of terminator cells (T).

	00	01	11	10
00	1	1	**1**	1
01	1	**1**	**0**	**1**
11	1	1	**1**	1
10	1	1	1	1

(a)

	00	01	11	10
00	1	1	1	1
01	1	1	1	**1**
11	**1**	1	**1**	**0**
10	1	1	1	**1**

(b)

	00	01	11	10
00	1	1	**1**	1
01	1	1	1	1
11	1	1	**1**	1
10	1	**1**	**0**	**1**

(c)

	00	01	11	10
00	1	1	1	1
01	1	**1**	1	1
11	**1**	**0**	**1**	1
10	1	**1**	1	1

(d)

TABLE I: Karnaugh maps for alternating represser cells.

alternating repressers in this way. Note that we implement each represser cell using inverters and a single NAND gate.

3) Terminators: The functionality of the mPPUF depends only on the time of the first $0 \rightarrow 1$ signal transition at each output of the final level of cells. Therefore, we use a terminator cell for each output to terminate all but the first $0 \rightarrow 1$ transition in order to increase the stability of the inputs to the arbiters. A terminator cell is simply a k-input OR gate that shares the same signal for all of its inputs.

An OR gate with high-enough k switches once for the first $0 \rightarrow 1$ transition and never again. This is due to the following three reasons: (i) an OR gate is 0 if and only if all of its inputs are 0; (ii) the represser gates drive all of the outputs to be 1 a majority of the time; and (iii) due to PV, the delay from each input of the OR gate to the output is different. Because any 0 signals exist for so short a time, it is very unlikely that for high-enough k's (in practice, $k = 4$) a 0 remains as input long enough to drive the OR gate's output to 0.

VI. Operation

In order to match two mPPUFs A and B, a subset of gates on A are aged to match the delays of corresponding gates on B, another subset of gates on B are aged to match the corresponding gates on A, and the remaining gates are disabled on both mPPUFs. In this section, we explain the mPPUF matching process in detail.

A. Aging and Disabling Strategy

In order to determine which gates on mPPUF A can be aged to match those on B and vice versa, consider the delay and aging models presented in Equations 1 and 2, respectively. A gate g_A on A can be aged to match its corresponding gate g_B on B if and only if: (i) g_A originally has lower delay than g_B; and (ii) the V_{th} of g_A can be increased enough by static aging to make its delay equal to that of g_B. The required V_{th} shift and aging time can be calculated from the two equations.

To allow for fast and low-energy device aging, we augment our mPPUF architecture by adding individual gate input control. The target gates on the mPPUFs can be aged by their respective desired amounts by supplying the gates with their maximally aging inputs. Note that it is beneficial for the two mPPUFs to match the greatest subset of gates possible, in order to maximize simulation complexity, achieve the desired statistical properties, and decrease the probability of coincidence from an unwelcome third party.

The final step in the physical matching process is for all the remaining unmatched gates to be disabled. In order to facilitate gate disabling, we add simple disabling logic (a single multiplexer) to the output of each gate that freezes the output to 1 or 0 if the gate is to be disabled.

B. Energy Optimization

In order to minimize energy spent in transmission while maintaining security, we propose a method to eliminate outputs in three phases. This procedure is based on the observations that the most beneficial outputs in terms of security are (i) those with close to 50% probability of being 0 (1) and (ii) those that are not easily predicted by others. This information can be obtained by applying a small set of input vectors after matching and conducting fast, low-cost statistical analysis.

In the first phase, we combine outputs that are often 0 (1) into a single output which is 0 (1) only if all of those outputs are 0 (1). This can be done with minimal hardware overhead using an additional level of OR and AND gates. Next, we eliminate those outputs that are 0 (1) with probability $P > |0.5 - \delta|$ for specified δ. Finally, we eliminate outputs that can be predicted by other outputs with certainty greater than a specified threshold (Section VIII). In other words, we transmit a maximal independent set of outputs that are not often 0 (1).

Furthermore, we propose two different schemes for arbitrating between output signals and clocks, one which minimizes transmission energy and one which maximizes security. In the first scheme, all outputs are arbitrated against a single clock signal. In this case, the response to a challenge consists of all

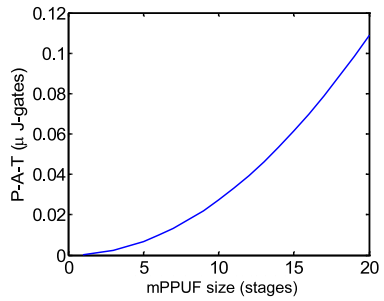


Fig. 3: Power-area-time product vs. number of stages for a mPPUF of width $w = 128$, 0.13 μm technology and $V_{dd} = 0.75V$.

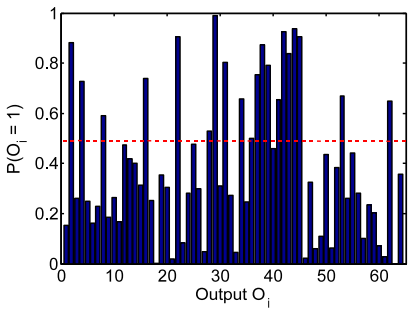


Fig. 4: Probability that an output bit of a matched mPPUF equals 1, shown for a subset of outputs for $w = 1024$ and $s = 7$. The red dashed line depicts the ideal case, where $P(O_i = 1) = 0.5$.

outputs not eliminated by the aforementioned procedure and is computed in a single clock cycle. The chosen clock period is the one that maximizes output entropy, which can again be determined using statistical analysis. In the second scheme, we increase output entropy at the cost of time and energy by executing the challenge multiple times with different clocks and selecting each output such that its entropy is maximized.

We present the power-area-time (P-A-T) product for computation vs. # of stages (s), for a mPPUF with configuration $w = 128$, $b = 2$, and $r = 1$, in Figure 3. We see that even for a very large mPPUF with $s = 20$ (36k equivalent gates), P-A-T is very low (0.11 μJ-gates). This is largely due to the fact that the entire computation is done in a single cycle, with critical path delay 2.6 ns, total power consumption 1.2 mW, and total energy 3.1 pJ. To compare with the most efficient AES encryption design presented in [28], we conducted these simulations using older 0.13 μm technology and $V_{dd} = 0.75V$. We find that the mPPUF improves P-A-T by a factor of over 100x and energy by three orders of magnitude.

VII. Protocols

We present the following protocols that can be achieved using PPUF matching: entity authentication and public key storage and communication. In the subsequent sections, we refer to two mPPUF holders Alice and Bob, each of whom knows the gate delay characteristics of the other's mPPUF, which are assumed public. Furthermore, we refer to $E_A(m)$ and $E_B(m)$ as the encryption functions provided by Alice's and Bob's mPPUFs, respectively, on a message m. Note that after Alice and Bob match their PPUFs, $E_A(m) = E_B(m)$.

Algorithm 1 Entity Authentication

1: Alice and Bob match their mPPUFs.
2: Alice chooses a message p and computes $E_A(p)$.
3: Alice sends p to Bob.
4: Bob computes $E_B(p)$.
5: Bob sends $E_B(p)$ to Alice.
6: Alice compares $E_A(p)$ to $E_B(p)$. If and only if $E_A(p) = E_B(p)$, Alice authenticates Bob.

Algorithm 2 Public Key Communication

1: Alice and Bob match their mPPUFs.
2: Bob chooses a message p and computes $E_B(p)$.
3: Bob computes $M = m \oplus E_B(p)$.
4: Bob sends p and M to Alice.
5: Alice computes $E_A(p)$.
6: Alice computes $m = E_A(p) \oplus M$.

A. Entity Authentication

Entity authentication, presented in Algorithm 1, is a basic cryptographic protocol and perhaps the most intuitive use of PUFs in general. This protocol relies on the properties that (i) only a mPPUF matched with Alice's can produce the same response to the same challenge and (ii) Bob's mPPUF is the only one that matches Alice's.

B. Public Key Storage and Communication

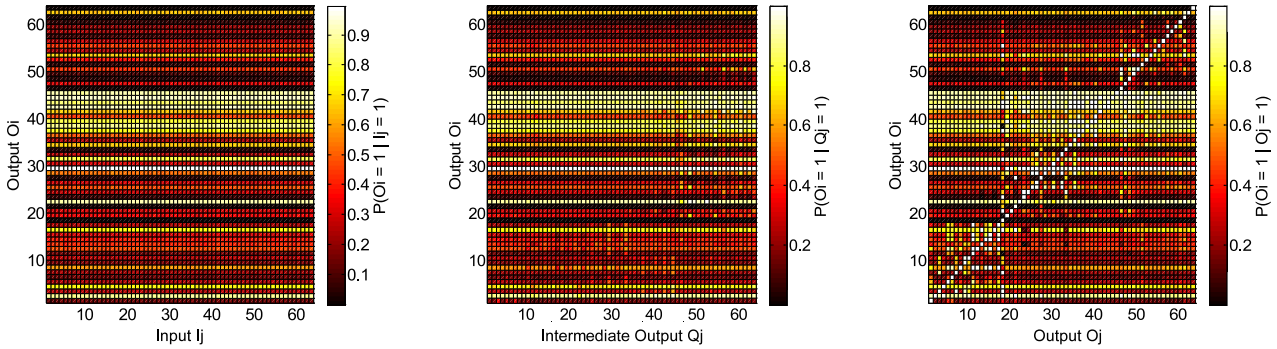
Public key storage and communication is at the foundation of public key cryptography. In public key communication, Bob sends Alice a message m such that Alice can read it but no other party can learn any new information about m (other than its encrypted value). The protocol proceeds as described in Algorithm 2. Because $E_A(p) = E_B(p)$ if and only if Alice and Bob have matched their mPPUFs, only Alice is able to extract the full original message m.

Alternatively, for public key storage, Alice does not match her mPPUF with any other mPPUF (i.e. all gates remain enabled), computes $M = m \oplus E_A(p)$, and stores M and p. To decrypt the message, Alice computes $m = E_A(p) \oplus M$.

VIII. Attacks

In this section, we identify potential security attacks against mPPUFs. We conducted comprehensive simulations using mPPUFs with $s = 7$, $w = 1024$ and 10,000 input vectors.

1) A priori guessing: In this attack, the attacker tries to predict each output O_i by observing a set of previous responses and calculating $P(O_i = 1)$ for all i. Figure 4 shows these probabilities for one representative set of mPPUFs after matching. We see that some outputs have very high or very low probability of being 1; in fact, some outputs are disabled and therefore completely predictable. However, there are many outputs that have $P(O_i \approx 0.5)$, the ideal case. Protocols can choose to use only these most unpredictable outputs.



(a) (b) (c)

Fig. 5: Conditional probability between output bits O_i and (a) input bits I_j, (b) intermediate output bits Q_j, and (c) other output bits O_j, shown for a subset of inputs and outputs for $w = 1024$ and $s = 7$. A value of 0.5 indicates complete unpredictability.

2) Correlated input-based prediction: An attacker can also try to combine previous observations with knowledge of the current inputs to augment his chances of correctly guessing outputs. For example, if the attacker observes that output i is 1 a large majority of the time that input j is 1, and if the current input j is 1, then he can guess that output i will be 1 with higher likelihood. We present the probability $P(O_i = 1|I_j = 1)$ for a subset of outputs O and inputs I in Figure 5a.

3) Correlated output- and intermediate output-based prediction: Similarly, an attacker can try to use partial computations to assist in output guessing by: (a) computing intermediate outputs Q at a height $h' < h$ and using correlations between those and the final outputs to determine a more likely guess; or (b) computing only some subset of the outputs and leveraging output-output correlations to better guess the remaining ones. We present the probabilities $P(O_i = 1|Q_j = 1)$ and $P(O_i = 1|O_j = 1)$ for a subset of outputs O and intermediate outputs Q in Figures 5b and 5c, respectively.

IX. Conclusion

We have presented the matched public PUF, an ultra low power cryptographic primitive that enables security protocols such as authentication and public key communication that require only a single clock cycle energy consumption for all participating parties. To the best of our knowledge, the mPPUF is the first such primitive, and the first PUF implementation that leverages device aging to facilitate self- and group-reconfigurable public keys. Simulation results show resiliency to a wide variety of security attacks and energy requirements that are three orders of magnitude less than those of recently proposed hardware implementations of AES encryption.

Acknowledgment

This work was supported in part by the NSF under awards CNS-0958369, CNS-1059435, and CCF-0926127.

References

[1] F. Dabiri and M. Potkonjak, "Hardware aging-based software metering," *DATE*, pp. 260–465, 2009.
[2] M. Nelson et al., "SVD-based ghost circuitry detection," *IH*, 2009.
[3] F. Koushanfar and M. Potkonjak, "CAD-based security, cryptography, and digital rights management," *DAC*, pp. 268–269, 2007.
[4] Y. Alkabani et al., "Remote activation of ICs for piracy prevention and digital right management," *ICCAD*, pp. 674–677, 2007.
[5] A. R. Brown, V. Huard, and A. Asenov, "Statistical simulation of progressive NBTI degradation in a 45-nm technology pMOSFET," *T-ED*, vol. 57, no. 9, pp. 2320–2323, 2010.
[6] S. Smith et al., "Comparison of measurement techniques for linewidth metrology on advanced photomasks," *SM*, vol. 22, no. 1, pp. 72–79, 2009.
[7] J. S. J. Wong, P. Sedcole, and P. Y. K. Cheung, "Self-Characterization of combinatorial circuit delays in FPGAs," *ICFPT*, pp. 245–251, 2007.
[8] A. Keshavarzi, et al., "Measurements and modeling of intrinsic fluctuations in MOSFET threshold voltage," *ISLPED*, pp. 26–29, 2005.
[9] S. Wei et al., "Gate-level characterization: foundations and hardware security applications," *DAC*, pp. 222–227, 2010.
[10] M. Potkonjak et al., "Hardware Trojan horse detection using gate-level characterization," *DAC*, pp. 688–693, 2009.
[11] S. Wei, S. Meguerdichian, and M. Potkonjak, "Malicious circuitry detection using thermal conditioning," *IEEE TIFS*, 2011.
[12] A. Vahdatpour et al., "A gate level sensor network for integrated circuits temperature monitoring," *IEEE Sensors*, pp. 652–655, 2010.
[13] B. Cline, K. Chopra, D. Blaauw, and Y. Cao, "Analysis and modeling of CD variation for statistical static timing," *ICCAD*, pp. 60–66, 2006.
[14] M. A. Alam and S. Mahapatra, "A comprehensive model of PMOS NBTI degradation," *Microelectronics Reliability*, vol. 45, pp. 71–81, 2005.
[15] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions," *Science*, vol. 297, no. 5589, pp. 2026–2030, 2002.
[16] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," *CCS*, pp. 148–160, 2002.
[17] N. Beckmann and M. Potkonjak, "Hardware-based public-key cryptography with public physically unclonable functions," *IH*, pp. 206–220, 2009.
[18] U. Rührmair, "SIMPL systems, or: can we design cryptographic hardware without secret key information?" *SOFSEM*, vol. 6543, pp. 26–45, 2011.
[19] M. Potkonjak et al., "Differential public physically unclonable functions: architecture and applications," *DAC*, 2011.
[20] M. Potkonjak, S. Meguerdichian, and J. L. Wong, "Trusted sensors and remote sensing," *IEEE Sensors*, pp. 1104–1107, 2010.
[21] M. Majzoobi, F. Koushanfar, and M. Potkonjak, "Lightweight secure PUFs," *ICCAD*, pp. 670–673, 2008.
[22] S. Meguerdichian and M. Potkonjak, "Device aging-based physically unclonable functions," *DAC*, 2011.
[23] S. Meguerdichian, "Device aging-based PPUFs: architecture and protocols," M.S. thesis, Computer Science Dept., UCLA, 2011.
[24] D. Markovic et al., "Ultralow-power design in near-threshold region," *Proceedings of the IEEE*, vol. 98, no. 2, pp. 237–252, 2010.
[25] S. Sarangi et al., "VARIUS: a model of process variation and resulting timing errors for microarchitects." *SM*, vol. 21, no. 1, pp. 3–13, 2008.
[26] A. Asenov, "Random dopant induced threshold voltage lowering and fluctuations in sub-0.1 um MOSFETs: a 3-D atomistic simulation study," *T-ED*, vol. 45, no. 12, pp. 2505–2513, 1998.
[27] S. Chakravarthi et al., "A comprehensive framework for predictive modeling of negative bias temperature instability," *IRPS*, pp. 273–282, 2004.
[28] T. Good and M. Benaissa, "692-nW advanced encryption standard (AES) on a 0.13-μm CMOS," *VLSI*, vol. 18, no. 12, pp. 1753–1757, 2010.