Nano-PPUF: A Memristor-based Security Primitive

Jeyavijayan Rajendran†, Garrett S. Rose‡, Ramesh Karri†, and Miodrag Potkonjak§
† Polytechnic Institute of New York University, Brooklyn, NY, USA ‡ Air Force Research Laboratories, Rome, NY, USA
§ University of California, Los Angeles, CA, USA
Email: jrajen01@students.poly.edu, garrett.rose@rl.af.mil, rkarri@poly.edu, miodrag@cs.ucla.edu

***Abstract*—CMOS devices have been used to build hardware security primitives such as physical unclonable functions. Since CMOS devices are relatively easy to model and simulate, CMOS-based security primitives are increasingly prone to modeling attacks. We propose memristor-based Public Physical Unclonable Functions (nano-PPUFs); they have complex models that are difficult to simulate. We leverage sneak path currents, process variations, and computationally intensive SPICE models as features to build the nano-PPUF. With just a few hundreds of memristors, we construct a time-bounded authentication protocol that will take several years for an attacker to compromise.**

I. Introduction

Physical Unclonable Functions (PUFs) leverage random physical disorders in the IC design process to produce unique responses (outputs) upon the application of challenges (inputs) [1]. PUFs are hardware-based secret key based mechanisms. Their unique responses are used to i) secure software execution on a processor [2], ii) active metering of ICs to prevent over production [3], iii) device authentication, iv) trusted configuration of FPGAs [4], and v) encrypted storage [2].

The strength of a PUF lies in the assumption that the attacker cannot model the PUF circuit and thereby cannot predict the response for a given challenge. However, researchers were recently able to model a PUF and predict its responses using machine learning techniques [5].

A. Public PUFs

A variant called the Public PUF (PPUF) has been proposed [6]. Unlike a PUF, the simulation models of a PPUF circuit are publicly available. While an attacker can simulate the PPUF for a given challenge to obtain a response, the simulation time is too large (several years) compared to the time it takes to apply the challenge and obtain the response on the PUF (a few seconds). The PPUFs [6] use several hundreds of thousands of XOR/XNOR gates to achieve this level of security (i.e. several orders of magnitude difference between simulation time and execution time).

B. Nano-PPUF

We proposed a PPUF using nanoscale memristor device (the nano-PUF). We use a memristor-based crossbar to build nano-PPUFs as shown in Figure 1. Inputs (challenges) to the PPUF are applied from the left, and outputs (responses) from the PPUF are observed at the bottom. The triangles in Figure 1 represent points where the internal voltages can be observed. The following features of memristors are the basis for their use in a nano-PPUF.

- Its simulation model is complex.
- It is bidirectional device and sensitive to process variations.
- For the same size of a CMOS PPUF [6], a very large number of memristors can be packed on chip, forcing the attacker to put more effort to generate the corresponding response on a simulator.
- Memristors are CMOS-compatible.

C. Time-bounded authentication

Consider the situation where Bob requests Alice to authenticate him. Bob has the nano-PPUF. The time-bounded authentication as shown in Figure 1 works as follows:

1. Alice obtains the simulation model of Bob's nano-PPUF from the public registry.
2. Alice randomly picks a challenge (V_{in1}, V_{in2}, etc.) and sends it to Bob.
3. Bob applies this challenge to the nano-PPUF that is in his possession. He sends the response (V_{out1}, V_{out2}, etc.) to Alice. This takes an order of several nanoseconds (ignoring the network delay for now).
4. Alice picks a subsection of Bob's nano-PPUF, say the 2×2 section at the left-bottom of the crossbar in Figure 1. She then requests from Bob the voltages at the boundary of the selected section (e.g., V_C, V_D, V_E, V_F) shown as red points in Figure 2.
5. Bob measures these voltages and sends them to Alice.
6. Alice simulates the selected section for the given challenge and the boundary conditions (V_C, V_D, V_E, V_F), and obtains V_{out1} and V_{out2}. The time to simulate this smaller crossbar is much less than the time taken to simulate the entire nano-PPUF.
7. If Bob's response matches Alice's response (obtained by simulation), then she authenticates him.

Now consider Chris, an attacker, can request Alice to authenticate him as Bob. He can perform any of the following

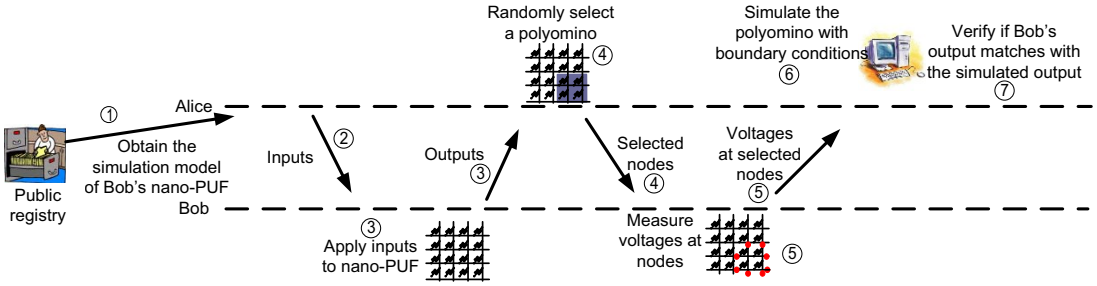

Figure 1. Protocol for time-bounded authentication using nano-PPUF.

978-0-7695-4767-1/12 $26.00 © 2012 IEEE
DOI 10.1109/ISVLSI.2012.40

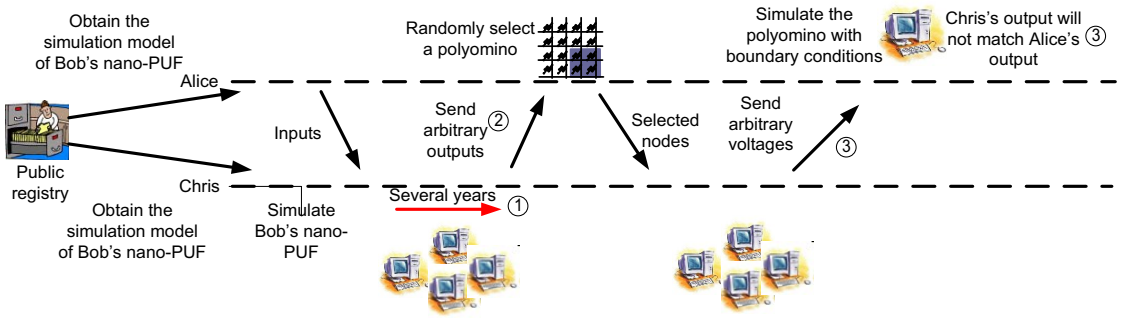

Figure 2. Chris, an attacker, can perform any of the two attacks to fake Bob's identity: 1 simulate Bob's nano-PPUF model which will take several years, 2 pick a random subsection (a 2×2 square) and simulate it. Then, tweak the outputs and boundary conditions to match the simulation results of that subsection.

attacks as shown in Figure 2.

1. Chris can obtain the simulation model of Bob's nano-PPUF from the public registry, simulate with the inputs provided by Alice and then send back the outputs. However, Chris will require more time to simulate than Bob and so Alice will reject Chris's reply.[1]
2. Instead of simulating the entire system, Chris could choose a subsection and Alice might coincidentally choose the same subsection for her simulation. Then Chris' output will match Alice's simulation results. However, the probability of this coincidence is low if the number of possible subsections is numerically large (>10^{10}).

While there was an earlier work on using crossbars as PUF, they do not produce any simulation results on PUF's cryptographic properties [16]. In this work, we support our claims using simulation results. Furthermore, nano-devices are previously used for trusted sensing [17].

II. PRELIMINARIES

A. Memristor-based crossbars

An N×N crossbar consists of two sets of N wires running perpendicular to each other. A resistive device such as a memristor is grown at the crosspoints.

Memristors are resistive devices fabricated by placing a metal oxide (such as TiO_{2-X}) layer with oxygen vacancies, on a perfect metal oxide (TiO_2) layer, and sandwiching them between metal electrodes [7]. When a large positive voltage difference is applied across the device over a period of time, its memristance will change to M_{on} (ON state). If a negative voltage difference is applied over a period of time, the memristance will change to M_{off} (OFF state). In between M_{on} and M_{off} states, memristors can exhibit several additional states which change as a function of voltage and time.

The memristive device fabricated by HP [7] is considered in this paper. Lateral dimensions of this memristor are 50 nm × 50 nm and the oxide thickness is 50 nm. M_{off} is 121 MΩ and M_{on} is 121 KΩ. The high M_{off} to M_{on} ratio (≈1000) and CMOS compatibility features facilitate in building crossbar based structures [7].

When M_{off}>> M_{on}, the relation between the memristance of the device M(φ(t)) and flux φ(t) at time *t* is given as [9] :

$$M(\varphi(t)) = \frac{\rho_{off}}{LW}\sqrt{\left(D^2 - 2\eta\frac{\rho_{on}}{\rho_{off}}\varphi(t)\mu\right)} \qquad (1)$$

where ρ_{on} and ρ_{off} are the resistivities of the TiO_2 layers with and without oxygen vacancies, respectively. L, W, and D are the length, width, and thickness parameters of the memristor, respectively. The parameter η (±1) is the polarity of the applied voltage signal and μ is the mobility of the ions.

B. Effect of variations

The memristive behavior is primarily affected by (i)the thickness of the oxide layers or device thickness, *D*, (ii) the lateral dimensions of the device, *L* and *W* and (iii) the doping concentration of the oxygen vacancies in the TiO_{2-X}layer, μ.

From Equation 1, variations in the lateral dimensions (L and W) are inversely proportional to the memristance values. The effect of process variation in D on memristance is highly non-linear; memristors are more sensitive to variability near M_{on} than near M_{off}. The memristance values differ by several orders of magnitude (10^3) due to variation. This feature is exploited to produce unique hardware signatures.

C. Developing simulation models

Accurate simulation models of the nano-PPUF are necessary for time bounded authentication. This model can be developed only if all memristors in the crossbars are characterized in terms of their physical properties such as length, width, thickness, and doping concentration. This characterization can be done either by directly measuring the physical parameters of the individual memristors using sophisticated microscopes [11], or characteristics such as delay and power and then solve systems of equations [12].

III. DESIGN OF A NANO-PPUF

The strength of the nano-PPUF relies on 1) the simulation of the nano-PPUF by an attacker (Chris) is computationally infeasible and 2) there are a large number of subsections of the nano-PPUF that Alice can pick for her simulation.

A. Memristor-based crossbars

Computational complexity in terms of the simulation of a nano-PPUF depends upon 1) the complexity of the memristor model and 2) the complexity in simulating a crossbar. Memristor simulation models tend to be complex because they include computationally intensive operations such as the integration and square root for each device as shown in

[1]Our analysis indicates that if the crossbar size is reasonably high, then the simulation time will be several years.

Equation 1. The simulation of a crossbar is hard as there is an exponential number of paths from an input to an output if the devices at the crosspoints are bidirectional[2].

There are two kinds of paths in a crossbar. In a direct path, current flowing from an input (row) to an output (column) is the function of the resistance of the device at the crosspoint of that input and output. In sneak path, the current flowing from an input to an output is a function of resistance of devices at other crosspoints in the crossbar.

If Chris wants to simulate and obtain an output for a given input, he has to evaluate the current flowing through all the sneak paths. Increasing the size of the crossbar exponentially increases the number of paths and thereby Chris's simulation time is exponentially increased. As there is an exponential number of sneak paths and N outputs in a crossbar, the simulation time for Chris is given by $N \times 2^N \times t_{memristor,\ sim}$.

The size of the crossbar also affects Bob's delay in obtaining an output. On applying an input, the current flows in parallel through the sneak paths and reaches the output. As there are N^2 devices in the crossbar, the execution time, t_{exe}, is $N^2 \times Delay_{memristor}$. Notice that increasing the crossbar size exponentially increases Chris's effort but increases Bob's execution time quadratically.

B. Polyominoes

Alice needs to select a section of the nano-PPUF to perform her simulation to authenticate Bob. The constraints on this selection are:

1. The possible number of sections should be exponential. If the number of possible sections is small, Chris can simulate all possible combinations apriori.
2. The size of the section should be small enough for Alice to simulate in real time.
3. At least one of the outputs should be a part of this section as Alice has to verify Bob's output with her simulated output.

To satisfy these constraints, we use polyomino shapes. The total number of possible polyomino shapes with M cells is exponential as given by the Equation 2 [13]:

$$\#\ of M-ominoes = \frac{c\lambda^M}{M}, \qquad (2)$$

where λ and c are 4.0626 and 0.3169, respectively. Alice will choose any of the polyominoes with at least one of N outputs of the crossbar is in the polyomino. The number of possible polyominoes of size M in an N×N crossbar is $\frac{c\lambda^M}{M}\times$ N.

Figure 3 shows six tetraominoes (a polyomino with four cells). Alice can choose any of the tetraominoes and set the boundary conditions for the selected tetraomino. Each memristor is a cell in the tetraomino. At least one of the polyomino cells lies in the edge of the crossbar so that Alice can validate Bob's outputs. In a crossbar of size N×N, there are N outputs and Alice can choose any one of them.

The simulation time of the polyomino depends on the size of the polyomino and the simulation time of each memristor. With a polyomino of size M, there are M memristors connected linearly. Hence, the simulation time of a N-omino, $t_{M\text{-}omino,sim}$, is M times the simulation time of a memristor, $t_{memristor,sim}$.

[2]Conventional memory design uses a reduced simulation model of the devices in the sneak paths for corner analysis where every device is assumed to have the same resistance value. An attacker cannot use a reduced model because the crypto-properties are based on the fact that the resistance values are different due to process variations.

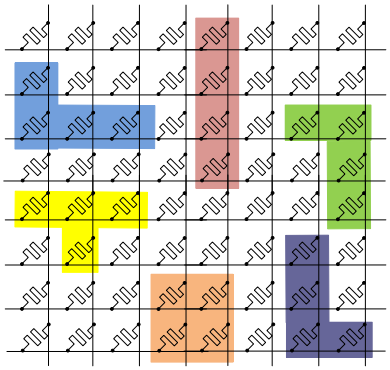

Figure 3: A 8×8 nano-PPUF. Alice can choose any of the six tetra-ominoes for her simulation to authenticate Bob.

Now consider the impact of polyomino size and crossbar size on the different parties involved in an authentication session. Bob needs to apply the inputs to the system and get the response as fast as possible. Decreasing the crossbar size helps Bob obtain the output faster but the crossbar must remain large enough to protect from Chris, the attacker. Alice needs to simulate a single polyomino from a set of exponential polyominoes to obtain a response. Alice's simulation speed can be improved linearly by decreasing the polyomino size.

Chris can do one of two things. First, he can simulate the design for all possible challenges and obtain the response. But for large crossbars the simulation becomes exponentially harder. Alternatively, he can randomly select a polyomino, simulate it, and reply back to Alice as fast as possible, hoping that Alice will pick the same polyomino. Increasing the size of the polyomino exponentially increases total number of possible polyominoes, reducing the probability of this coincidence.

C. CMOS voltage sensors

In order for Alice to simulate the selected polyomino, she needs the voltages at the boundaries of that polyomino. Bob' nano-PPUF should have accurate voltage sensors, to measure the voltages similar to those in [10].

IV. Results

A. Experimental setup

Simulation models of the memristor device fabricated in [7] were developed based on Equation 1 using LTSpice. These devices switch from ON to OFF or from OFF to ON at the rate of 10^{-9} seconds [7]. To analyze the effect of process variation, we varied the thickness of the devices. We applied all possible challenges for a 4×4 crossbar on 100 different crossbars to determine the uniqueness of responses. The simulation time for polyominoes and crossbars were obtained using LTSpice.

B. Polyomino size-M

Increasing the polyomino size exponentially increases the number of possible polyominoes but also linearly increases the simulation time for Alice. Figure 4 shows the number of possible polyominoes and the simulation times for different sizes of polyominoes in a 20×20 memristor-based crossbar. The left-side Y-axis is log scaled while the right-side Y-axis is linearly scaled. One can infer from the figure that even for a small size of the polyomino such as 20, the number of possible polyominoes is more than a billion. Chris does not know which

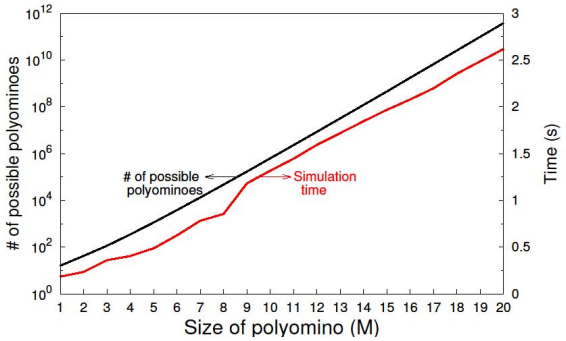

Figure 4. Number of possible polyominoes and the simulation times for different sizes of polyominoes in a 20×20 memristor-based crossbar.

of these billion polyominoes that Alice will select for her simulation. With a polyomino size of 20, the simulation time for Alice is around 2.5 seconds which is feasible to perform real time authentication.

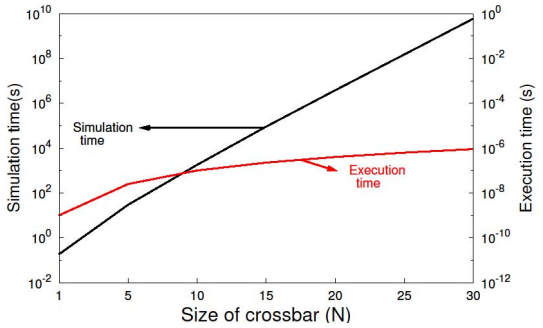

Figure 5. Simulation and execution times for different sizes of memristor-based crossbar.

C. *Size of the crossbar-N*

Increasing the crossbar size exponentially increases the simulation time for Chris but also quadratically increases the execution time. As shown in Figure 5, even for a 20×20 crossbar, the simulation time is several tens of years (10^9) seconds. However, Bob, who has the nano-PPUF, can obtain the response within several microseconds. The difference between the simulation and execution time is 10^{15} seconds and is the basis for a secure time-bounded authentication protocol.

D. *Unique responses*

To analyze the proposed nano-PPUF for producing unique IDs, we used two different metrics: i) the intra-crossbar Hamming distance (Hamming distance between the responses from a crossbar upon application of two different challenges which differ by 1-bit) and ii) the inter-crossbar Hamming distance (Hamming distance between the responses from two different crossbars upon application of the same challenge). Ideally, both of these metrics should be 50%.

The second and third rows of Table 1 show the inter- and intra-crossbar Hamming distances, respectively. It can be seen that the average is 49% in both the cases.

V. CONCLUSION

While conventional VLSI design tries to eliminate sneak paths, reduce effects of process variations, and build compact simulation models, we leveraged these factors to build a security primitive. Even with just a few hundreds of memristors, we were able to build a PPUF which will take several years for an attacker to simulate. Furthermore, the developed nano-PPUF is more robust to changes in temperature then CMOS devices [14] and is rad-hard as memristors characteristics are not affected by radiation [15].

Table 1. Average Inter and Intra Hamming distances for different amount variation in memristor's oxide thickness.

Amount of variation	1%	2%	3%	4%	5%
Inter HD	50%	49%	49%	49%	49%
Intra HD	49%	49%	49%	49%	49%

VI. ACKNOWLEDGEMENTS

Acknowledgements: Received and cleared for public release by AFRL on May 11, 2012, case number 88ABW-2012-2779. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of AFRL or its contractors.

VII. REFERENCES

[1] B. Gassend, et al., "Silicon physical random functions," ACM Conf. on Computer and Communications Security, pp. 148–160, 2002.

[2] G. E. Suh, et al., "Design and implementation of the AEGIS single-chip secure processor using physical random functions," ACM SIGARCH Computer Architecture News, vol. 33, pp. 25–36, May 2005.

[3] Y. Alkabani and F. Koushanfar, "Active control and digital rights management of integrated circuit IP cores," Int'l Conf. on Compilers, architectures and synthesis for embedded systems, pp. 227–234, 2008.

[4] J. Guajardo, et al., "Physical unclonable functions and public-key crypto for FPGA IP protection," Int'l Conf. on Field Programmable Logic and Applications, Aug. 2007, pp. 189–195.

[5] U. Ruhrmair, et al., "Modeling attacks on physical unclonable functions," ACM Conf. on Computer and Communications Security, pp. 237–249,2010.

[6] N. Beckmann and M. Potkonjak, "Hardware-basedpublic-key cryptography with public physically unclonablefunctions," Information Hiding, pp. 206–220, 2009.

[7] D. B. Strukov, et al., "How we found the Missing Memristor," Nature, vol. 453, pp. 80–83, 2008.

[8] Y. N. Joglekar and S. J. Wolf, "The elusive memristor: properties of basic electrical circuits," European Journal of Physics, vol. 30, no. 4, pp. 661–675, 2009.

[9] J. Rajendran, et al., "Approach to Tolerate Process Related Variations inMemristor-Based Applications," Intl Conf. on VLSI Design, pp. 18–23, 2011.

[10] A. Mason, et al., "A mixed-voltage sensor readout circuit with on-chip calibration and built-in self-test," IEEE Sensors Journal, vol. 7, no. 9, pp. 1225–1232, sept. 2007.

[11] P. Friedberg, et al., "Modeling within-die spatial correlation effects for process-design co-optimization," Intl Symp. on Quality of Electronic Design, pp. 516–521,2005.

[12] S. Wei, et al., "Gate-level characterization: Foundations and hardware security applications," ACM/IEEE Design Automation Conf., pp. 222–227, 2010.

[13] I. Jensen and A. J. Guttmann, "Statistics of lattice animals (polyominoes) and polygons," Journal of Physics A: Mathematical and General, vol. 33, pp. L257–L263, 2000.

[14] J. P. Strachan, et al., "The switching location of a bipolar memristor: chemical, thermal and structural mapping," IOP Nanotechnology, vol. 22, no. 25, 2011.

[15] W. Tong, et al., "Radiation hardness of TiO_2 memristive junctions," IEEE Trans. on Nuclear Science, vol. 57 no. 3, pp. 1640–1643, 2010.

[16] U. Rührmair, et al., "Applications of High-Capacity Crossbar Memories in Cryptography" IEEE T. Nano. vol. 10, no. 3, pp. 489-498, 2011.

[17] J. B. Wendt, M. Potkonjak, "Nanotechnology-Based Trusted Remote Sensing", IEEE Sensors, pp. 1213-1216, October 2011