PUF designed with Resistive RAM and Ternary States

Bertrand Cambou
Northern Arizona University
2112 s Huffer Lane - PO Box: 5621
Flagstaff AZ 86011
1(928)5237824
Bertrand.cambou@nau.edu

Marius Orlowski
Virginia Institute of Technology
614 Whittemore Hall
Blacksburg, VA 24061, USA
1(540)2313297
Marius@vt.edu

ABSTRACT
The designs of Physically Unclonable Functions (PUFs) described in this paper are based on Resistive RAMs incorporating ternary states with the objective to generate predictable Challenge-Response-Pairs (CRPs). The ternary states, the "Xs", allow the blanking of all cells that are not characterized as consistently capable to generate stable and easy to read "1s" or "0s" PUF challenges. Experimental data extracted from Cu/TaOx/Pt Resistive RAM samples confirms that such a method can generate CRPs having error rates below 8 ppm useable for highly secure hardware authentication. Random Number Generators (RNG) can also be enhanced by the same ternary architecture.

General Terms
Algorithms, Design, Experimentation, Security.

Keywords
Hardware Authentication, Software & Data protection, Encryption, Secure Memory, Physically Unclonnable Functions, Resistive RAM.

1. Introduction & Background information

1.1 Physically Unclonable Functions (PUFs)
PUF are strengthening the level of security of emerging authentication methods, and this as part of a set of cryptographic primitives. PUFs act as a virtual or "finger prints" of the hardware by providing unique signatures during the authentication process. PUFs exploit intrinsic manufacturing variations, which are introduced during the fabrication of the devices such as critical dimensions, doping level of semiconducting layers, and threshold voltages (References [1], and [2]) making each device unique, and identifiable from each other. The underlying mechanism of PUF is the implementation of a generator of a large number of Challenge (i.e. input) Response (i.e. output) Pairs (CRPs). The generation of CRP's has to be reproducible, predictable, and easy to recognize during the authentication process. The challenges can be stored in a secure server. Once deployed during the authentication cycles, the PUFs are queried with Responses. The authentication is positive when the rate of matching Pairs is high. Weak PUFs produce marginal rates of matching Pairs, while with strong PUFs the rates are close to 100%. Other criteria to judge the quality of a PUF are the size of the Challenges, and the robustness of the CRPs matching when subject to temperature, voltage, EMI, aging, and other factors. Randomness, uniqueness, and secrecy are bound to make PUFs extremely hard to extract for unwelcomed users, and easy to use for secure authentication. Memory based PUFs are mainstream, in particular when generated with SRAM.

1.2 Memory based PUF
Table 1 summarizes previously reported methods to generate PUFs out of SRAMs, DRAMs, ReRAMs, MRAMs and Flash

Table 1: PUF CRP generation from memories.

Memory type	*Example of PUF patterns*	*Comments*
6T - SRAM	Random Flip of the 6T cell: start as a "0" or a "1" after power up	Mainstream but not secure
DRAM	Constant discharge of the capacitors, then measure: Get a "0" or a "1"	Create weak PUFs
Resistive RAM	Variations of the value of the Rmin's after programming: Define a "0" or a "1"	High potential
Magnetic RAM	Variation of the Rmax's after programming: Define a "0" or a "1"	Unknown
Flash RAM	Partial programming of the cells, then measure: Get a "0" or a "1"	Create weak PUFs

Memory devices have physical natural parameters subject to manufacturing variations with threshold that can be used to determine how to program streams of "0s" and "1s", with equal probability, see Figure 1. When applied to a large number of cells this has the potential to create digital signatures exploitable as part of a PUF; references [3], [4], [5], [6], and [7]:

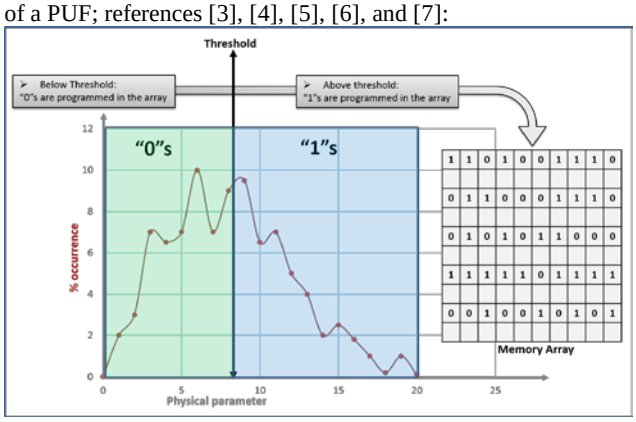


Figure 1: Physical variations for PUF challenges.

1.3 Security with memory based PUFs
One method is to generate PUF Challenges with memory devices and store them in a secure server. The Responses are generated by the same memory each time the devices are powered up. CRPs matching are checked during the authentication process by (or for) the secure server. Key figures of merit for PUFs are quantifying the level of protection of the secret information during Challenge-Response generation cycles, storage, as well as the error rate during the authentication process. Designing PUFs with Resistive RAMs is attractive because these memories operate extremely fast at low power making them hard to be hacked.

2. Creation of PUF CRP's with ternary states

2.1 Determination of ternary states

The cells close to the threshold as shown Figure 1 can flip one way or the other during Challenge generation cycles, and can drift in the opposite direction when subject to aging, temperature, voltage changes, or electromagnetic interferences. If the ratio of these marginal cells is too high this could result in large number of CRPs matching errors in the 5 to 20% range. Error detection and error correction algorithms can be effective for the generation of stronger PUFs, however they might expose the device to crypto-analysts.

The solution offered to this problem in this work (see Figure 2) is based on the identification of three type of cells, and this during the Challenge generation cycle: the ones that are solidly a "0" or a "1" far away from the threshold determining the difference between the two states (as shown Figure 1), with the remaining ones being given a ternary state "X".

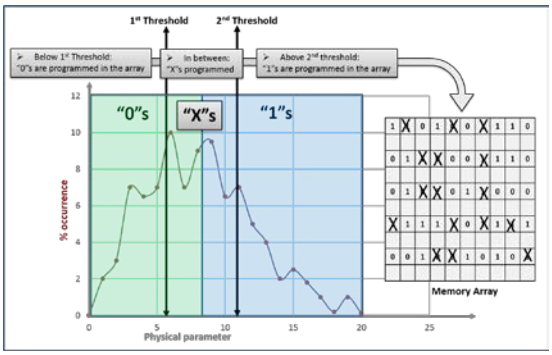


Figure 2: Determination of ternary states in a memory

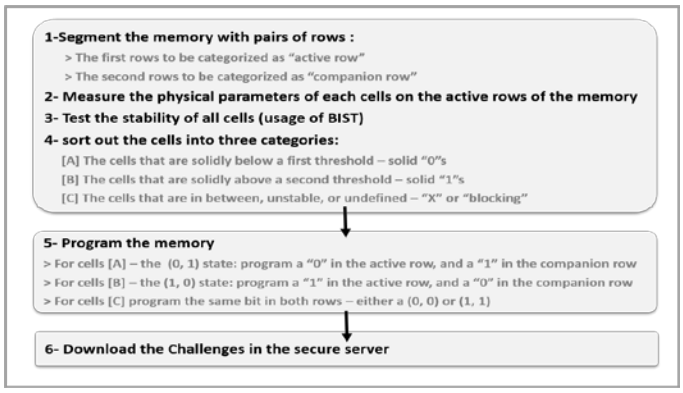
The main objective of the method is to increase the probability of a solid bits, "0" or "1", to stay stable and predictable with CRPs matching error rates in the part per million range (ppm). The cells to be blanked by an "X" state include the cells that are too close to the transition point, and the ones that are not reliable. Mainstream BIST (Built In Self-Test; reference [8]) modules as developed to test memory products can be implemented to sort out all marginal cells, and blanking them with an "X" state. Unlike prior arts (example reference [9]) there is no need for external circuitry to store the "X" state, the PUF memory itself can store the three states (0, 1, X), as described below in section 2-2 to 2-5. Ways to obtain ternary states in commercial memory products has been presented in prior work (Reference [10], [11]).

2.2 Algorithms to generate PUF Challenges

Figure 3 describes an example of algorithm that generate PUF challenges from a memory having ternary states which are submitted to a secure server for future authentication.

Step 1- configuration of the memory. The memories are segmented by pairs of rows (or columns). The cells of the first row of each pair, are the "active rows (or columns)". The cells of the second row (or column) of each pair, the "companion rows (or column)", are the ones where complementary information is stored to describe the three elementary states (0, 1, X).

Step 2- measurement of the physical parameters. The physical parameter should be measured in each cell of the active rows (or columns) for the purpose of the generation of PUF Challenges.



. Figure 3: Algorithm – Challenge Generation.

Step 3- test the stability of all cells. It is important at this point to sort out the solid cells versus the unstable ones, even if the testing process could be lengthy. As discussed section 2.1, BIST modules commonly used to reduce the cost of testing commercial memories can be also utilized for this purpose.

Step 4 – Sort out the cells into three categories. The determination of the status of all cells of the active rows, i.e. a "0", a "1" or a "X", shall be done at the end of the full testing of the memory. Any questionable cells should be classified as an "X".

Step 5- Programming of the memory. When a particular cell of an active row has been characterized as a solid "0", a "0" shall be programmed in this cell, and a "1" shall be programmed in the companion cell. In a similar way, a solid "1" shall trigger the programming of a "1" in the active row (or column), and a "0" in the companion row (or column). All remaining cells that do not yield solid bits shall be programmed with a ternary state, an "X", with both bits in the active and companion locations been identical, either a "0, 0" or a "1, 1". [It is possible to program only "0, 0"s, only "1, 1"s, and/or to alternate between the two].

Step6- Download the Challenges.

Option1: Binary PUF Challenge. The mapping of the "X"s is kept in the memory for future authentication, while the PUF Challenges transmitted to the secure server are purely binary, and can be used for multiple authentication of the hardware.

Option2: When the secure server can handle ternary data streams, Ternary PUF Challenges can be downloaded as is.

This PUF Challenge generation process is summarized Figure 4 with a simplified flow chart.

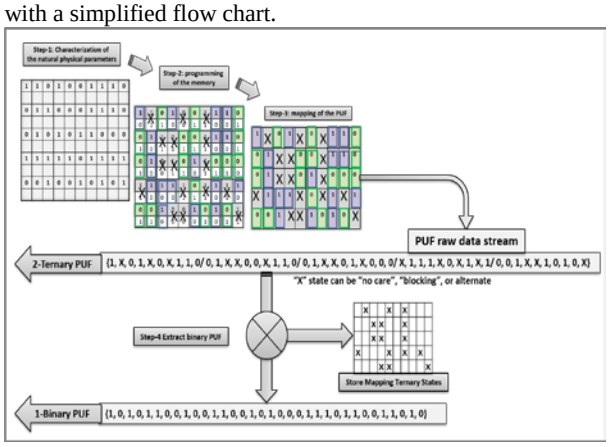


Figure 4: PUF Challenge generation – Flow chart

2.3 Algorithms of authentication

Figure 5 describe an example of algorithm for the authentication of PUFs with binary or ternary states. A secure server is providing PUF Challenges to the Response that are generated and stored in the memory after powering up the device.

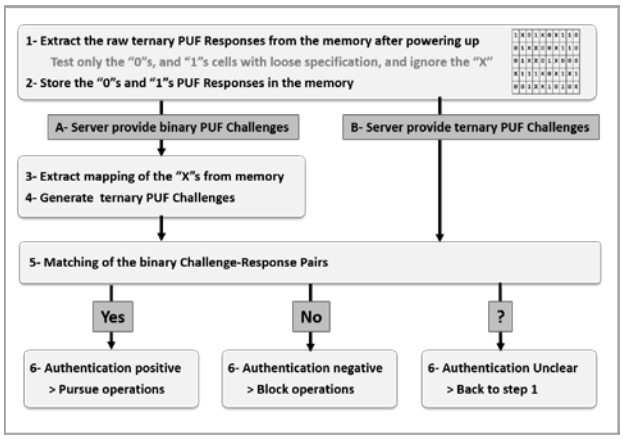


Figure 5: Algorithm – CRPs verification

Step1- Preparation of the PUF Responses after power-up. The memories generating PUFs patterns have to be erased to start fresh at every cycle when the device is powered up. After erase, the PUF Response are generated using an algorithm similar as the one described section 2-2 during Challenge generation but with looser criteria to guarantee low CRPs error rate.

Step 2- Storage of the ternary Responses. The Responses can be stored in the memory space that generated them, or any other memory arrays.

If the Challenges provided by the secure server are ternary, skip Step 3 & 4, and go straight to step 5.

Step3- Extract the mapping of the "X"s. In order to reconstruct a full pattern out of a Binary Challenge, the mapping of the position of the "X"s has to be extracted from the memory.

Step- 4 Generate PUF ternary Challenge. Combine the binary PUF Challenges with the mapping of the "Xs" to generate the PUF ternary Challenges.

Step5/6- authentication. The Responses generated by the PUF memory, and Challenges brought by the secure server are compared for the purpose of testing the level of matching between all CRPs, the "Xs" are ignored. Due to the strong quality of the binary data stream it is expected that the authentication signal resulting from the matching (or not) of the reference pattern with the one provided by the server will be strong.

When the Responses are stored in a RAM, these Responses have to be extracted from the RAM during the authentication cycles, and compared with the Challenges provided by the server. This could create security breaches. Storing the PUF Responses in a Content Addressable Memories capable to directly perform CRPs matching "in situ" can enhance security, Reference [12]. In the authentication process described Figure 6, the ternary data streams are transferred directly in a TCAM PUF memory for data matching.

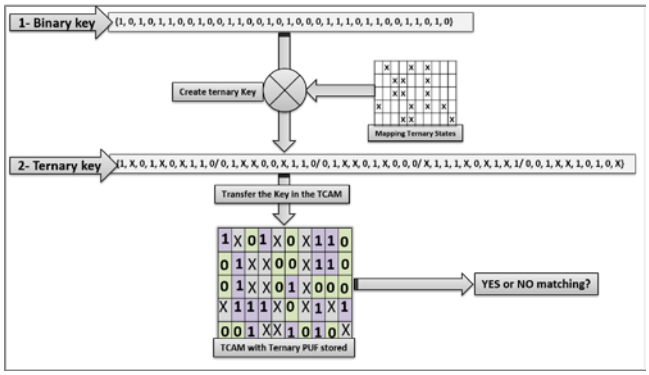


Figure 6: Authentication – Flow chart.

2.4 Enhancing cryptographic security

2.4.1 Entropy

If, for example, the PUF has N=128 bits of which 16 are potentially unstable, the level of certainty of a matching could fluctuate between 87% (112/128) and 100%. The portion of a memory that has to be set aside to generate PUF challenges, can be increased to keep the size of the challenges at a preferred level after blanking. For example 150 bits can be set aside to leave 128 clean bits after the blanking of 32 "X" bits. The important criteria is the stability of the "1s" and "0s", not the stability of the "Xs" because they are blanked, their actual physical states are irrelevant. This results in the enhancement of the entropy of the cryptographic system.

2.4.2 Ternary PUFs - Quaternary states

Prior art [reference 10] using ternary state (0, 1, random) can increase the entropy from 2^{128} to 3^{128} ($3^{128} \approx 2^{(128 \times 1.58)}$) with ternary logic. The reliability of the random states, the random is questionable, the entropy of 3^N is only a best case. This can be improved by using quaternary states "0, 1", "1, 0", "1, 1" and "0, 0" as described section 2.2. As shown Figure 7, the cells that are in the middle zone can be referred as the real "Xs", and be programmed with the state"0, 0". The remaining cells located in the buffer zone can be referred as "BX" and be programmed with the state "1, 1". Back on the previous example, the size of the PUF can be increased to 256 bits to leave 128 cells with ternary logic. As a result the ternary challenges (0, 1, X) can be predictable, and fully authenticable with an entropy of 3^N, N=128.

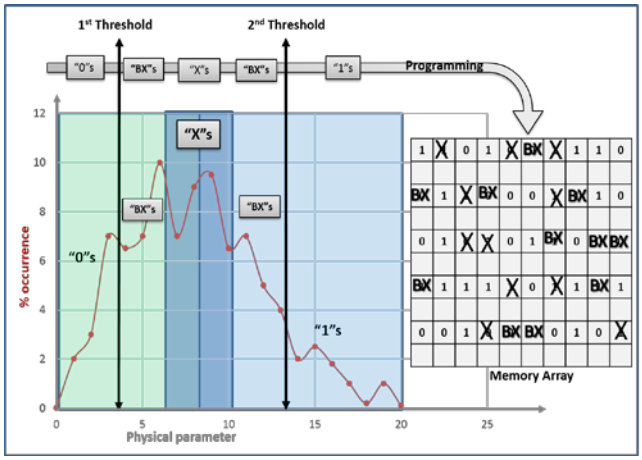


Figure 7: Usage of buffer states - ternary PUF.

2.4.3 Reduction of side channel attacks

All solutions have intrinsic weaknesses that could be exploited by crypto-analysts using side channel attacks such as Differential Power Analysis (DPA), or Electro Magnetic probes. “0”s and “1”s tend to generate different electric currents that are particularly visible during programming and reading cycles of the PUF and secure memories. In the method described in this paper, the “0s” are programmed as “0, 1”s, and the “1s” as “1, 0”s, so the measurements of electric currents generated during authentication cannot differentiate the “0s” and the “1s”.

2.4.4 Error Detection and Error Correction (ECC).

CRP matching quality can benefit from error detection/error correction during challenge generation cycles, or response operations, see Reference [14] and [15].The usage of ternary states or quaternary states to blank the bad cells can also act as an effective error correction step, further enhancing the strength of the PUF, thereby reducing the need for additional ECC.

2.4.5 Additional Encryption

The methods presented in this work can be combined with additional encryption operation. The work described in Reference [13] involving usage of ternary states (or blocking states) and Edit Distance algorithm is directly implementable to enhance security. Additional “X” states obtained from a cryptographic key or pin code can be inserted in the memory array to distort the stored pattern. These additional “X” can then be subtracted during the authentication cycle to restore the correct pattern.

2.4.6 Random Number Generation (RNG)

RNG can be a by-product of the PUF challenge generation process. After testing all cells of the memory array as described in step 3 of Section 2.2 it is possible to extract only the cells that are close to the transition point between “0s” and “1s”, blanking the rest “X”, and this combined with a Pseudo random selection of the order to pick these cells in the array for the Random Numbers. Such a method can yield random numbers at high data rate.Read times as fast as 1ns/bit have been reported on ReRAMs, so the method could reach the remarkable data rates of 1 Gbit/s.

3. PUF with ReRAM memories

3.1 Architecture

The suggested way to generate PUF CRPs, see Figure 8, is to characterize the natural variations in the resistivity of Vset, a parameter described section 3.2. The high values are re-programmed as “1s”, the low values “0s”, and the rest blanked “X”s. For a RAM architecture (a), the first “active” columns are the ones where “0”s and “1”s are tested during Challenges or Response generation. The second “companion” columns, are the ones where a”1s” are stored when “0s” are tested in the active column, the reverse when “1s” was tested, and an identical bit for an “X”. The CAM architecture (b) requires four columns per states: the first and the third columns are respectively the active and companion columns to store the Responses. The second and forth columns store the Challenges for authentication. All Challenges in word line 1 (ex: 1, X, 0, 1) can be checked for CRP matching at once. The logic function in Figure 8 is:
[{aXORb}OR{a’XORb’}] or **[{a XOR b}AND{a’XORb’}]**.

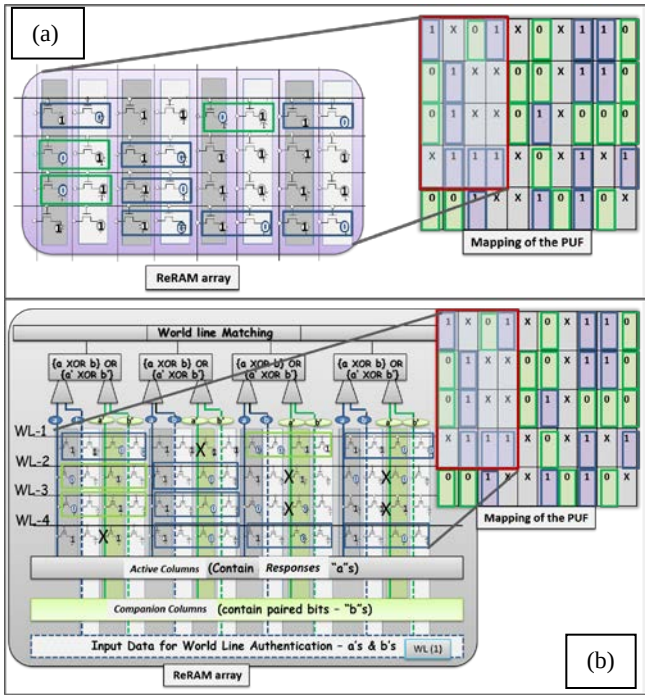


Figure 8: PUF CRPs with ReRAMs
(a) RAM (b) CAM

The architecture described above is relying on standard ReRAM arrays, and state machines. The subject of the following section is to experimentally verify that Vset variations can generate reliable ternary states (0, 1, X) as described in section 2.0 with CRPs allowing authentication cycles with low error rates.

3.2 Experimental data

3.2.1 Description of the ReRAM samples

In order to model realistic ReRAM PUF CRPs, Cu/TaOx/Pt resistive devices have been fabricated in a crossbar array on a thermally oxidized Si wafer, Reference [16]. Both metal electrodes and solid electrolyte were deposited by E-beam evaporation and patterned by lift-off technology. The oxygen-deficient TaOx, 16 nm thick, was deposited by evaporating TaOx pellets without O_2 injection to the evaporation chamber. The top Cu electrode runs perpendicularly to the bottom Pt electrode. The width of the metal lines varies between 1 μm and 35 μm. The device cross-section is shown in Figure 9 (a), (b) shows the microscopic top-view image of our fabricated array. A single Cu/TaOx/Pt switch relies on electrochemical formation and rupture of a Conductive Filament (CF) bridging the dielectric between the active Cu an inert Pt electrode. As shown in Figure 9 (c), there exists a minimum Vset voltage applied across the switch, at which a CF is being formed. When the voltage applied to the Cu electrode is pulsed or swept at a positive voltage, the current will remain substantially zero until a critical voltage Vset is reached, at which a Cu CF is formed connecting the Cu and Pt electrodes, and the cell switches from a high resistive state (HRS) characterized by Roff (1–900 MΩ) to a low resistive state (LRS) characterized by Ron (70–6000 Ω), yielding a ratio of Roff/Ron ≈ 10^3–10^7. When a negative voltage is applied to LRS state, CF ruptures at a critical voltage of Vreset and the cell switches from LRS to HRS state. The rupture of the CF is triggered by a critical

current Ireset = Vreset/Ron. Therefore, to ensure a successful set operation to logic state "1", the magnitude of the applied maximum voltage must be slightly larger than Vset voltage for the particular cell. Similarly, the operation for writing a logic "0" requires a reset maximum voltage slightly larger in magnitude than Vreset for a particular cell.

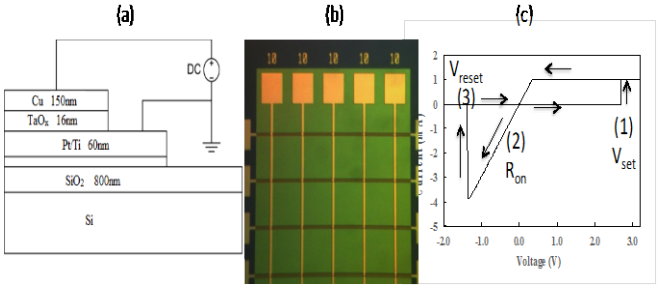


Figure 9 (a) Cross-section of the cell (b) Optical micrograph (c) Typical I-V characteristic with transition at positive Vset(1) and reset voltage Vreset(2). LRS state is characterized by Ron

The cell's switching parameters (Vset, Vreset$_t$, Ron, Roff) are subject to statistical variations. The variability of cell parameters degrades memory operation margins and the functional array design; nevertheless, a random variation of cell characteristic parameters lends itself to exploitation in security applications for PUF and TRNG generators. Regarding the immunity to physical attacks, ReRAM is very robust because the variations of characteristic parameters, such as Vset, Vreset, Ron, and Roff, stem from structural material properties and are atomic in nature. It is important to note that ReRAM variability is inherent not only in manufacturing variations but also in the electro-chemical ionic switching mechanisms of ReRAM device itself. Neither can a ReRAM device be probed by invasive techniques, as the atomic changes of the defect densities are hardly visible, even under high resolution transmission electron microscopy. In addition, ReRAM is less sensitive to side-channel attacks due to the low power characteristics during read cycles, as well as the attacks based on photon emission analysis; ReRAM is not emitting photons like hot carriers in MOSFET transistors of an SRAM or a floating gate MOSFET cell. ReRAM memory has also intrinsically higher density, faster access speed, and better energy efficiency than conventional memory technologies. Figure 10 shows the Ron and Roff resistance distributions of the samples.

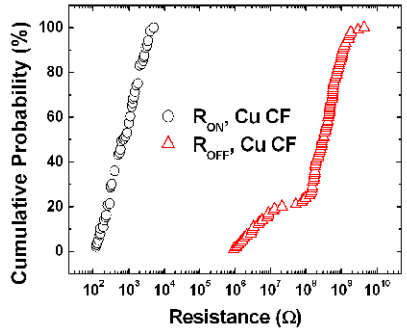


Figure 10 Cumulative distributions of Ron and Roff.

Both distributions could be used as a random source of variation for PUF challenge generation. The Roff distribution would have the advantage that is has larger variation, and also, it can minimize the parasitic voltage drop outside the active structure. From the preliminary data it was observed that the resistance shift under temperature change (0°C to 85°C) is less than 10% both for Ron and Roff.

3.2.2 PUF Challenge generation from Vset

In this paper we are studying the variations of the Vset voltage for the generation of PUF Challenge-Responses-Pairs. The Challenges are generated with stringent guard bands, and stored in a secure server. Figure 11 shows the cumulative Vset probability distribution within a typical sample of ReRAM memory array containing 10,000 cells.

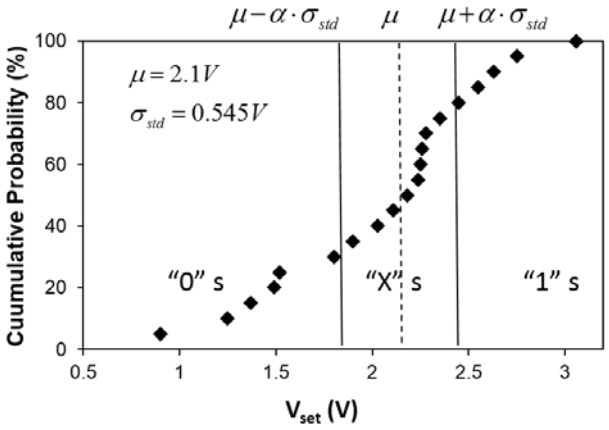


Figure 11: Cumulative Vset probability distribution for the entire array of cells. **μ** and **σ** of the distribution calculated.

The mean of this distribution is **μ**=2.1 V (indicated by the dashed line) and standard deviation is **σ**std=0.545 V. In Figure 9 right and left from the mean there a two lines at a distance of **α** x **σ**std from the mean, where **α** is a PUF design parameter. This voltage-dependent switching probability provides the source of randomness for the generation of PUF CRPs that is the subject of the analysis discussed below. All the cells in the array are first set to HRS. Then, by biasing the memory cells at Vset=μ, every cell has equal opportunity to be characterized as a "0" or a "1". In our approach the median **μ** and standard deviation **σ**set are being used to construct two dividers (1st and 2nd threshold, see Figure 9) for three types of cells.

During PUF Challenge generation, the cells are classified as "0", "X", and "1", when in the intervals [$V_{set\text{-}min}$, μ−ασ], [μ−ασ, μ+ασ], and [μ+ασ, $V_{set\text{-}max}$], respectively. μ−ασ corresponds to 1st threshold in and μ+ασ corresponds to 2nd threshold in Figure 9.

Those cells between minimum Vset (here 0.8V) and μ−ασ (1st threshold) will be denoted as "0" states, and kept as LRS on the active columns while the companion columns are programmed as HRS. Those between μ−ασ and μ+ασ are the "X" states, both the active and companion are programmed the same way with either LRS/LRS or HRS/HRS. Lastly, those between μ+ασ, maximum Vset (here 3.4 V) the "1" values are programmed as HRS while the companion columns are programmed LRS. The dimensionless parameter α is used in this work to run some statistical model

describing the populations of "0", "X", and"1" as well as the CRPs error rate. When α=0 the population of "X" is zero and the populations of "0" and "1" are equal. When $\alpha > 1$ more than 60% of the cells will be "X" states. It can also be seen that any $\alpha \neq 0$ will somewhat guard against undesired bit-flips lowering mismatch rates Challenge-Response. The larger α, the larger the safety margin against bit-flips, and the lower the CRPs error rate are expected to be. For large enough α any reasonable bit flip will be confined to "X" population while "0" and "1" Challenges are stables and predictable, and guarded against environmental (e.g. temperature) changes or (e.g. electromagnetic) interferences. Thus α can be used to strengthen the stability of the CRPs at the expense to blank a higher proportion of the cells as "X".

3.2.3 PUF Challenge-Response-Pair Matching

The PUF Challenges are generated by the ReRAM memory, as described section 3.2.2, based on Vset characterization, and then stored in a secure server. For binary logic, only the "0"s and the "1"s are stored in the server, the "Xs" are kept in the memory for further authentication as described section 2.0. The PUF Responses are generated with looser criteria by the same memory every time the system power up to be compared with the Challenges that were generated with tight margins, and stored in the server. The Responses are generated by testing only the "0" and the "1" population, ignoring the "Xs". For example a transition of 2.1 Volt can be set regarding Vset to determine the "0"s versus the "1"s of the Responses. CRP errors will only occurs when a "0" (as tested during Challenge generation within the interval [Vset-min, μ-$\alpha\sigma$]) is measured above 2.1 volt, or a "1" (as tested during Challenge generation within the interval [μ+$\alpha\sigma$, Vset-max]) is measured below 2.1 volt. The Responses are stored back in the memory together with the un-tested "Xs". During authentication, all binary CRPs are tested, comparing the patterns generated by the memory, the Response, against the previously generated patterns that are provided by the secure server, the Challenges. An acceptable design point of the PUF is the one where the cumulative variations in Vset will stay within the buffer zones to respectively determine the Challenges and the Responses, yielding to low CRPs error rate.

3.2.4 PUF CRP error rate

In order to study the robustness of the PUF method, and the CRPs error rate, we have characterized the Vset distribution for several individual ReRAM cells. For this characterization, see Figure 12, we have selected from the distribution a cell with a low Vset value ($Vset_t \approx 1V$) and a cell with a high Vset value ($Vset_t \approx 2.5V$). The cells have been subjected to repeated reset and set operations under the same conditions.

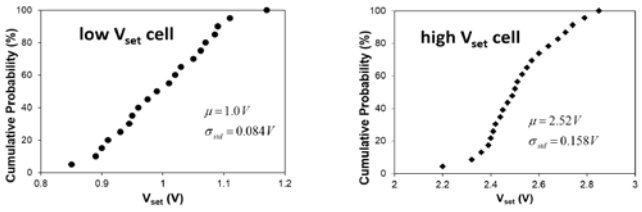


Figure 12: Cumulative Vset probability distributions for a single cell.
(a) Low Vset(≈1 V) cell. (b) High Vset(≈2.5 V) cell.

Vset distribution for the low Vset cell is centered around 1V, and its standard variation **σ**std=0.084 is smaller than the overall variation **σ**std=0.545 V of the array. For the high Vset cell we obtain **μ**=2.52 V, and **σ**std=0.158 V, also smaller than the array variations. In term of physics, one would expect that a cell that requires high electric field (i.e. high Vset) will display larger variation than a cell that switches at lower electric fields (i.e. low Vset). Of course, the standard deviations for the single cells will influence the choice of a parameter. Based on these results, the variation of each cells is plotted on Figure 13 as a function of the average Vset of these individual cells.

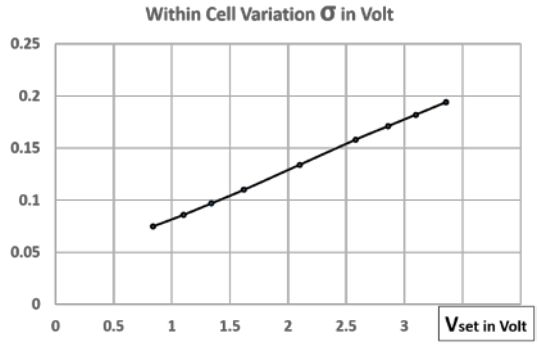


Figure 13: Within cell σ versus the average Vset

The statistical analysis based of these experimental data is presented Table 2. In this analysis α vary from 0.5 to 2.0, and the threshold point to generate the Responses is varying from 1.8 volt to 2.1 volt. The analysis assume normal distributions.

Table 2: Statistical analysis – PUF CRPs error rate

α	ασ	% of "0"s	Max For "0"s	Mean for "0"s σ0	μ0	% of "1" s	Min For "1"s	Mean for "1"s σ1	μ1	% of "X"s
0.5	0.27 V	37.5	1.83 V	0.106	1.62	37.5	2.37 V	0.162	2.58	25
1	0.54 V	16	1.56V	0.098	1.34	16	2.64 V	0.172	2.86	68
1.5	0.81 V	6.5	1.29 V	0.086	1.10	6.5	2.91 V	0.182	3.10	87
2	1.08 V	2	1.02 V	0.076	0.84	2	3.18 V	0.194	3.36	94

α	% of "0"s	Error rate CRP 0 read as a 1 Tipping @ 2.1V	Error rate CRP 0 read as a 1 Tipping @ 1.9V	Error rate CRP 0 read as 1 Tipping @ 1.8V	% of "1" s	Error rate CRP 1 read as a 0 Tipping @ 2.1V	Error Rate CRP 1 read as a 0 Tipping @ 1.9V	Error Rate CRP 1 read as a 0 Tipping @ 1.8V	% of "X"s
0.5	37.5	650ppm	80,000ppm	5%	37.5	35,000ppm	4,500ppm	200ppm	25
1	16	0	8ppm	1,000ppm	16	1,000ppm	8ppm	0.5ppm	68
1.5	6.5	0	0	0	6.5	75ppm	0.3ppm	0	87
2	2	0	0	0	2	0.3ppm	0	0	94

The impact of this error rate on the authentication cycle of a PUF stream of N bits can be calculated with Poisson equation. If P(n) is the probability to have n failures over N bits, p is the probability to have one CRP mismatch due to errors:

Eq (1) $$\mathbf{P(n) = \lambda^n/n!\ e^{-\lambda}}$$

Eq (2) $$\mathbf{\lambda = pN}$$

As shown on Table 2, the design point α=1.0 with a threshold of 1.9 volt has a CRPs error rate p=8ppm. With Poisson equation Eq(1), assuming N=128:

P(0)=99.2%; P(1)=0.794%; P(2)=30ppm; P(3)≈0.

At that level the probability that at least 126 bits are matching during the authentication cycle over 128 CRP candidates is almost certain. About 68% of the cells are blanked "X", and the rest can be used for CRP generation. For example in order to generate a PUF of 128 bits, the memory array involved need to be in the 1,000 bit range; 50% of the cells will be used as part of the companion cells.

3.2.5 Real Ternary PUF CRP

The experimental data extracted from the ReRAM samples can also be used to model a real ternary PUF CRP, as presented section 2.4.2. In this case an additional buffer zones has to be inserted between the states "0s", and "Xs", as well as between the "Xs" and the "1".For example the Challenges are extracted with the following intervals:

For "0s": Vset below 1.29V. This statiscally represents 6.5% of the cells, with a mean **μ**=1.1V, and **σ**=0.086V.

For "Xs": Vset between 1.9V and 2.0V, 7.5% of the cells, mean **μ**=1.95V, and σ=1.30V.

For "1s": Vset above 2.91 V, 6.5% of the cells, with a mean μ=3.1V, and σ=0.186V.

The rest, about 80% of the cells are blanked as "BX" . The Responses are set with a transition between "0s" and "Xs" at 1.45V, and a transition between "X"s and "1"s at 2.45V. With these assumptions the calculated error rates CRP are:

"0" read as a "X" = 2,000ppm
"X" read as a 0/1 = 8,000ppm
"1" read as a "X" = 10,000ppm

The worst case is "1s" reading as "Xs". This error rate is higher than what is expected from binary PUF CRP's, however considering that the entropy will be 3^N versus 2^N this is a good trade-off. Using Poisson equation with N=128, and p=10,000ppm(1%), P(n) the probability to have n errors is:

P(7)=300ppm; P(8)=50ppm; P(9)=7ppm; P(10)<1ppm.

The probability to have 118 CRPs match out of 128 is almost certain. The resulting entropy, $3^{118}=2\ 10^{56}$ is still greater than $2^{128}=3.5\ 10^{38}$.

3.2.6 Random Number Generator (RNG)

It is possible to generate Random Number with ReRAMs by reusing the method already presented to generate a PUF. One way is to program in advance the entire ReRAM arrays with "0"s and "1s"s based on Vset determination. The cells that are close to the transition point are kept for RNG, while the cells that are solidly "0s", "1s", and defectives can be blanked with an "X". Table 3 summarize the stastitical analysis of the method using the data presented Figure 11. With α=0.05 representing 4% of the cells, the ratio 1/0 for a single cell is expected to be in the 46% to 54% range for 50% of the cells and 54% to 46% for the other half.

Table 3: Statistical analysis – RNG

α	ασ	% of "0"s	% of "1" s	% of "X"s	"X" s read as a "0" Min	μ	Prob. flip For "X"s 0 > 1	"X"s read as a "1" Max	μ	Prob. flip For "X"s 1 > 0
0	0	50	50	0	Na	Na	Na	Na	Na	Na
0.05	0.027 V	48	48	4	2.072 V	2.087 V	46%	2.127 V	2.113 V	46%
0.1	0.054 V	46	46	8	2.046 V	2.076 V	43%	2.154 V	2.124 V	43%
0.15	0.081 V	44	44	12	2.019 V	2.060 V	38%	2.181 V	2.140 V	38%

It takes about 6,400 ReRAM cells to get 128 active cells usable for Random Number Generation. Considering that half of the cells will have 54% to be oriented in one direction, while the other half have only 46% to be oriented in the same direction, the mean distribution is centered at 64 bit in each state, like True Random Numbers. To improve the randomness the following methods can be implemented:

- Prepare a much larger number of cells candidates for RNG. The RNG can then continuously be done with new cells of random state.
- Pick the cells within the 128 candidates in an order that is random. This randomness will be then cumulative with the randomness of the state of each cell.
- Tighten the manufacturing variation of Vset increasing the ratio of cells at the threshold between "0s" and "1s". For example if the 4% of non-blanked cells turn into 10%, only 2,500 cells will be need for 128 bits, and the 46% to 54% ratio could be improved closer to 50%.

4. Future work

The approach to generate PUF on ReRAM arrays is promising based on the statistical analysis of the Cu/TaOx/Pt resistive devices. Next step is to generate large number of CRPs from these memory arrays, and experimentally extract the error rates to verify the statistical analysis. Characterization over temperature, and various biasing conditions has to be included in the study. The subsequent step will be to manufacture integrated solutions combining crypto-processors and ReRAM arrays.

5. Summary

The usage of ternary states, as presented in this work, can strengthen PUFs generated by memory devices, blanking all marginal states, delivering solid hardware cryptographic keys, and low CRPs matching errors during authentication cycles. The characterization of Cu/TaOx/Pt resistive ReRAM samples was performed to analyze the value of this method. In particular the variation of Vset, a parameter controlling the programming of the cells, has the potential to generate strong PUF CRPs with matching error rates in the 8ppm range. Resistive RAM is an attractive memory technology for designing secure applications, PUFs, and RNGs. It is low power, fast, compact, and less sensitive to side channel attack than flash memory.

6. Acknowledgement

Our thanks to Prof. Lionel Torres from the University of Montpelier, and Prof David Naccache from ENS Ulm for providing stimulating thoughts in chip security and cryptography.

7. References

[1] David. Naccache and Patrice. Frémanteau; Aug. 1992; Unforgeable identification device, identification device reader and method of identification; *Patent US5434917.*

[2] Ravikanth Pappu, Ben. Recht, Jason Taylor, and Neil Gershenfield; 20 Sept 2002; Physical one-way functions; *Science. Vol 297 No5589 pp2026-2030.*

[3] Pravin Prabhu, Ameen Akel, Laura M. Grupp, Wing-Kei S. Yu, G. Edward Suh, Edwin Kan, and Steven Swanson; June 2011; Extracting Device Fingerprints from Flash Memory by Exploiting Physical Variations; *4th international conference on Trust and trustworthy computing.*

[4] Daniel E. Holcomb, Wayne P. Burleson, Kevin Fu; Nov 2008; Power-up SRAM state as an Identifying Fingerprint and Source of True Random Numbers; *IEEE Transactions on Computers, vol 57, No 11.*

[5] Todd A. Christensen, John E Sheets II; Oct. 30, 2012; Implementing Physically Unclonable Function (PUF)

utilizing EDRAM memory cell capacitance variation*; Patent No.: US 8,300,450 B2;Assignee IBM.*.

[6] Xiaochun Zhu, Steven Millendorf, Xu Guo, David M. Jacobson, Kangho Lee, Seung H. Kang, Matthew M. Nowak, Daha Fazla; March 2015; Physically Unclonable Function based on resistivity of magnetoresistive random-access memory magnetic tunnel junctions; *Patents. US 2015/0071432 A1*.

[7] Elena I. Vatajelu, Giorgio Di Natale, Mario Barbareschi, Lionel Torres, Marco Indaco, and Paolo Prinetto; July 2015; STT-MRAM-Based PUF Architecture exploiting Magnetic Tunnel Junction Fabrication-Induced Variability; *ACM transactions*.

[8] Anuj Gupta, May 2005, Implementing Generic BIST for testing Kilo-Bit Memories; *Master Thesis No-6030402 Deemed University Patiala India.*

[9] Dai Yamamoto, Kazuo Sakiyama, Kazuo Ohto, and Masahiko Itoh; 2011; Uniqueness Enhancement of PUF Responses Based on the Locations of Random Outputting RS Latches; *Cryptographic Hardware and Embedded Systems – CHES 2011 Lecture Notes in Computer Science Volume 6917, pp 390-406.*

[10] Bertrand Cambou; June 24, 2015; Memory circuits using a blocking state; *US patent Application No: 22728483*.

[11] Bertrand. Cambou; June2, 2015; ReRAM architectures for secure systems; *US Application No 62/169957*.

[12] Bertrand Cambou, Neal Burger, Mourad El Baraji; May, 2014; Apparatus system, and method for matching patterns with an ultra-fast check engine, *US patent No 8,717,794B2*.

[13] Bertrand. Cambou; Jul 16, 2015; Multi-factor authentication using a combined secure pattern; *US patent Application No 22938751*.

[14] Christian Krutzik; Jan 2015; Solid state drive Physical Unclonable Function erase verification device and method; *US patent application publication US 2015/0007337 A1*.

[15] Dominik Merli, Frederic Stumpf, Georg Sigl; 2013; Protecting PUF Error Correction by Codeword Masking; *IACR Cryptography, e-print archive 2013: 334.*

[16] Gargi Ghosh and Marius Orlowski; 2015; Write and Erase Threshold Voltage Interdependence in Resistive Switching Memory Cells; *IEEE transactions on Electron Devices, 62(9), pp. 2850-2857, 2015*

Bertrand Cambou

Professor of Practice at Northern Arizona University. Dr. Cambou primary research interests are in cyber-security, and how to apply nanotechnologies to strengthen hardware security. Previously he worked as CEO in Silicon Valley in nanotechnologies where his organization won a contract with IARPA with applications related to quantum cryptography. He worked in the smartcard industry at Gemplus (now Gemalto), and in the POS/secure payment industry at Ingenico. He spent 15 years at Motorola Semiconductor (now NXP-Freescale), 5 years as CTO; he was named “Distinguished Innovator” and scientific advisor of the BOD. He is the author and co-author of 35 patents in microelectronics and cybersecurity; hold a Doctorate degree from Paris-South University, France, and an Engineering degree from Supelec, France.

Marius Orlowski

Dr. Orlowski has been a Professor and the Virginia Microelectronics Consortium Chair with Virginia Polytechnic Institute and State University, Blacksburg, VA, USA, since 2008, where he continues his research in semiconductor device physics, and microelectronics technology. For over 26 years he worked in the semiconductor industry, first at Siemens, then at Motorola SPS, now Freescale/NXP. He received a Master Innovator and Distinguished Innovator awards at Motorola, IEEE Fellow award in 1998 and a Fulbright Fellow award in 2014. He is the author and co-author of 109 patents, including 80 US patents, and has over 240 scientific publications. Received a PhD (1981), MS (1977), and BS (1974) from Tübingen University, Germany.