A Non-Volatile Memory Based Physically Unclonable Function without Helper Data

Wenjie Che, Jim Plusquellic
ECE Dept., University of New Mexico
Albuquerque, NM, USA
wjche@unm.edu, jimp@ece.unm.edu

Swarup Bhunia
EECS Dept., Case Western Reserve University,
Cleveland, Ohio, USA
skb21@case.edu

***Abstract*—Stability across environmental variations such as temperature and voltage, is critically important for Physically Unclonable Functions (PUFs). Nearly all existing PUF systems to date need a mechanism to deal with "bit flips" when exact regeneration of the bitstring is required, e.g., for cryptographic applications. Error correction (ECC) and error avoidance schemes have been proposed but both of these require helper data to be stored for the regeneration process. Unfortunately, helper data adds time and area overhead to the PUF system and provides opportunities for adversaries to reverse engineer the secret bitstring. In this paper, we propose a non-volatile memory-based (NVM) PUF that is able to avoid bit flips without requiring any type of helper data. A voltage-to-digital converter technique is described for digitizing the analog entropy source and a robust median-finding algorithm is proposed as the reprograming strategy. Analysis on published experimental data is presented to demonstrate the practicability of our proposed strategy. We describe the technique in the context of emerging nano-devices, in particular, resistive random access memory (Memristor) cells, but the methodology is applicable to any type of NVM including Flash.**

***Index Terms*—Physically Unclonable Functions, Helper data, Memristor**

I. Introduction

Physically Unclonable Functions (PUFs) are emerging as an alternative to programming embedded secret keys in ROMs and non-volatile memories (NVMs) in integrated circuits (ICs). PUFs extract entropy from variations in the physical and electrical properties of ICs, that are unique to each IC, as a means of generating secrets. These secrets can be used in various security applications including device identification, authentication, metering, remote activation and encryption [1].

Applications such as encryption require precise regeneration of the secret bitstring, possible under different environmental conditions. This requirement presents challenges for PUFs (in contrast to secrets that are programmed into non-volatile memories or NVMs) because the entropy source leveraged by PUFs is analog in nature and hence can be significantly impacted by changes in temperature and voltage (TV) noise. Moreover, distinguishing subtle differences in the entropy source is further challenged by measurement noise in many cases.

The most popular approach to dealing with these challenges is to extract error correcting information from the secret bitstring once it is generated for the first time during the enrollment process, that is later used to correct errors which occur during regeneration [2]-[5]. The error correction information is stored in reliable, digital storage, e.g., in an on-chip NVM or an off-chip storage device. A second thresholding-based technique 'avoids' bit flips by being selective regarding which components of the entropy source can be compared reliably to generate a bit [6]. However, thresholding techniques also require helper data that indicate which comparisons are reliable.

In this paper, we propose a NVM-based PUF implementation that does not require helper data for regeneration. The entropy leveraged in our scheme is the manufacturing variations that occur in the transconductance (or resistive) characteristics of the NVM cells. The enrollment process measures and digitizes these variations and then 'programs' the NVM cells with the random bitstring that is produced. Therefore, the full reliability of the NVM is used to preserve the bitstring across power cycles and under varying TV conditions, which allows regeneration processes to extract it without suffering bit-flip errors.

The enrollment process is carried out in a special manner. First, elements of the PUF's entropy source are stimulated and digitized using an on-chip measurement structure that is capable of providing 'soft information'. Soft information implies that the magnitude of the analog variations are digitized into multi-bit, e.g., 8-bit, digital values. A distribution is then constructed using these digital values and a median-finding algorithm is used to partition the population into two segements (with an equal number of elements in each segment). NVM cells with digital values in the lower half of the distribution are programmed with a '0' while those in the upper portion are programmed with a '1'.

Our proposed NVM PUF, by its very nature, defeats one of the stated advantages of PUFs, i.e., PUFs eliminate the cost of including NVM on the chip and the need to store the secret bitstring in digital form. However, the NVM PUF does preserve the basic premise of a PUF, namely, that the secret is derived from manufacturing variations and is not programmed (or even known) by the manufacturer as is true in the traditional use of NVM. The real benefit of our proposed scheme is in the use of NVM cells as both a source of entropy and a means of eliminating public 'helper data'. We recognize that storing the secret in NVM memory represents a vulnerability and may disqualify the NVM PUF for high security applications that need to protect against invasive probing attacks. However, the small footprint and the guarantee of high reliability of the NVM scheme make it attractive for other, lower security, small form factor, applications.

In this paper, we describe the NVM PUF enrollment and

regeneration processes in the context of Memristor devices. Published data on within-die variations in Memristor arrays is leveraged to show proof-of-concept and to guide the design of on-chip measurement infrastructures which are capable of measuring and digitizing NVM cell resistance variations.

We present the background in Section II and provide an overview of the proposed PUF design in Section III. This is followed by Section IV which introduces the memristor device and its characteristics. The proposed voltage-to-digital converter and the PUF architecture are presented in Section V, and Section VI evalutes the proposed strategy using measured data from published literature. Section VII concludes the paper.

II. Background

The feasibility of building a Memristor-based PUF is recently discussed in [7]. The authors utilized a weak write mechanism which leverages the resulting unpredictable logic states to implement a PUF. The evaluations only focused on the uniqueness metric without consideration for the stability metric of the PUF. The work in [8] proposed a Memristor-based PUF structure that is based on the randomness of the resulting programming state of two cells in series that occurs after a reset operation. Garrett S. Rose et al. proposed a Memristor-based PUF that leverages the write time variability of the Memristor device [9]-[10]. It is implemented by choosing the actual SET time close to the minimum SET time so that the percentage of the output logic '0' or logic '1' will be each nearly 50%. Another PUF that integrates a Memristor device into the conventional RO-PUF structure is proposed in [11]. Variations in high state and low state resistance after a programming operation is used as the entropy source. The authors demonstrate that the randomness in the resistance values increases the number of CRPs of conventional RO-PUFs.

In contrast to previous work, the primary goal of our proposed PUF is to eliminate bit flips and the need for any type of helper data. To realize this goal, we exploit the characteristic that the bimodal resistance profiles of Memristor devices are widely separated and therefore, the membership of a specific device in either profile can be determined reliably. Other contributions of this paper include 1) a stimulus circuit and an on-chip voltage-to-digital converter (VDC) scheme for obtaining soft information on the resistance characteristics of the NVM cells and 2) a median-finding algorithm that is robust to non-Gaussian resistance distribution profiles.

III. Overview

As indicated in the Introduction, the elimination of helper data is a major benefit of the proposed NVM PUF. Fig. 1 illustrates the mechanism by which this is accomplished. The resistance distribution shown along the bottom left illustrates the randomness that exists in the resistance of the Memristor cells programmed in the low resistance state (LRS). This distribution represents the entropy source for the NVM PUF. As described in the next section, the analog resistance values are digitized and the median of this distribution is determined. The histograms shown in the upper left depict the profiles after a selected set of NVM cells are reprogrammed into the high resistance state (HRS). The large gap between these profiles ensures that subsequent regenerations always make the correct decision regarding the profile to which a given NVM cell belongs. This is a key distinguishing feature of the NVM PUF. All other PUFs (to our knowledge) must operate on the LRS distribution for regeneration. Unfortunately, varying TV conditions and measurement noise change the LRS distribution profile, making it impossible to generate the same bitstring without helper data.

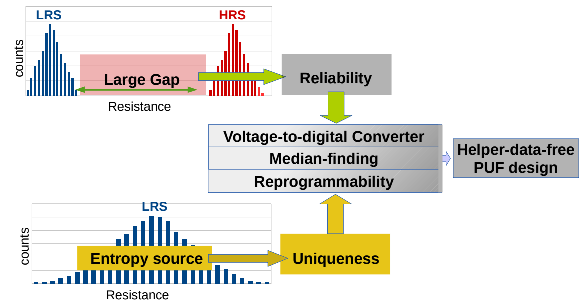

Fig. 1 Overall concept of the NVM PUF design

Fig. 2 further illustrates the enrollment process in more detail. The flow starts with programming all the memristor cells to the low resistance state, by using the “write-operation” of memristors which is described in the next section. A voltage-to-digital converter (VDC) is then responsible for digitizing the sensed voltage drop across each memristor cell to a digital value between 0 to 128. All these digitized values are stored in an SRAM array that is cell-to-cell mapped to the memristor arrays. In order to create a histogram, a state machine is used to count the number of instances for each digitized value (from 0 to 128). This is achieved by storing the counts of each digital value into a second 129-cell on-chip SRAM, whose addresses represent the counters storing the corresponding digital values. Then a state machine is utilized to find the median digital value

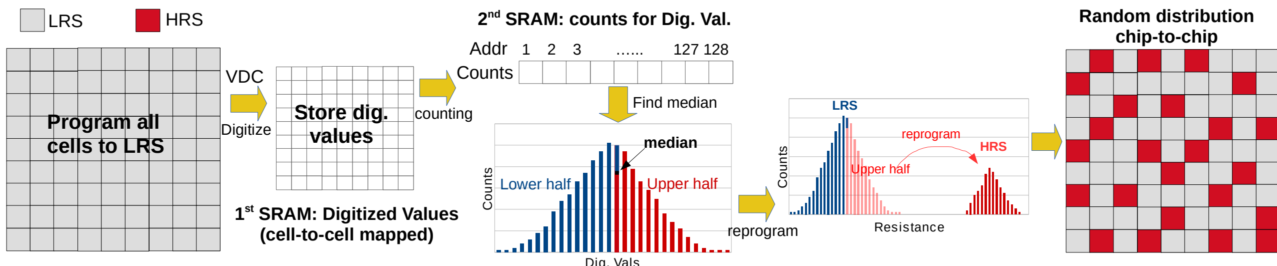

Fig. 2 Overflow of the enrollment strategy for the proposed NVM-based PUF

of the profile by adding up the counter values from low to high addresses. The median value is then recorded and used as a divider that determines which memristor cells are going to be reprogrammed to the high resistance state (those with larger values than the median). All cells will be randomly split into two equal-numbered groups, LRS and HRS profiles, after the reprogramming procedure. The random variations of LRS determine how the resistance profiles for a specific NVM array distributes, with each generating a unique pattern as shown on the far right in Fig. 2. Therefore the generated bitstring between two different chips will be unique.

IV. Memristor Devices and Key Features

Memristors have become a mainstream research topic because of their advantages as novel memory primitives over conventional memory technologies including static RAMs (SRAMs) and Flash memories. For example, Memristors have intrinsically higher density, faster access speed and better energy efficiency [12]. Memristors are also classified as a NVM technology, in which special write operations can be used to configure cells into one of two (or more) resistance states.

A Memristor is an electrical switch that is able to retain internal resistance states according to its history of applied voltage and current [13]. The different resistance states can be sensed to generate logic '0's and '1's. Memristor write and read operations are implemented by applying write or read voltage pulse patterns. Different patterns are used for the reading and writing operations.

Fig. 3 shows the structure of a Memristor cell and the mechanisms used for read and write operations. As shown in Fig. 3(a), a Memristor cell is composed of two electrodes and a metal oxide doping layer sandwiched between them. The length of the doping region *w* will be extended to the maximum length of D when the dopants are fully constructed (doped), and reduced to 0 when dopants are completely destroyed (undoped). The resistances of the completely doped region and undoped region can be represented by R_{on} and R_{off} respectively. Equation (1) gives an expression for the overall resistance as a function of the doping extent *w*.

$$R(w)=R_{on}\cdot\left(\frac{w}{D}\right)+R_{off}\cdot\left(1-\frac{w}{D}\right) \quad (1)$$

The doping behavior can be controlled by applying voltage pulses of the appropriate magnitude and duration as shown in Fig. 3(b). The change in the doping characteristic of the Memristor cell changes its resistance characteristics. This is depicted in the figure and labeled as LRS for low resistance state and HRS for a high resistance state, which corresponds to a logic '1' and '0', respectively. To write a logic '1', V_{in} should generate a positive square voltage pattern with magnitude V_A and time duration T_{W1}. To ensure a successful write, the magnitude of V_A must be larger than the threshold write voltage of $V_{th,w1}$ and the duration T_{W1} must be longer than $T_{th,\,w1}$. Similarly, the operation for writing a logic '0' requires a negative write voltage $-V_A$ with duration of at least $T_{th,\,w0}$. Note that some memristors cannot be configured properly after manufacture until being conditioned with a larger formation voltage V_f [15]. We assume memristors used in our enrollment algorithm have been "formed" for normal configurations. To perform a read operation, a voltage pulse pattern is required that is composed of a negative pulse followed by a positive pulse with equal magnitude and duration [14]. The negative pulse is used to detect the current internal state but it also perturbs the doping state of the cell. The subsequent positive pulse is designed to re-generate the doping conditions and corresponding resistivity of the original state. This pattern of read pulses is illustrated in Fig. 3(c), which also shows when the corresponding output value is available for reading, in particular, the intervals t_1-t_2 for read '1' and t_4-t_5 for read '0'.

Fig. 4 shows the histogram of the HRS and LRS variations extracted from a 1600 Memristor devices (40*40 nanocrossbar array) [16]. The spread in the distributions illustrates that the resistance of a Memristor cell after a write operation varies considerably, and is due to process variations and voltage variations over the Δt of the write operation. This characteristic makes it challenging to use Memristor cells for a PUF in cases where the resistance variations within either of the two states

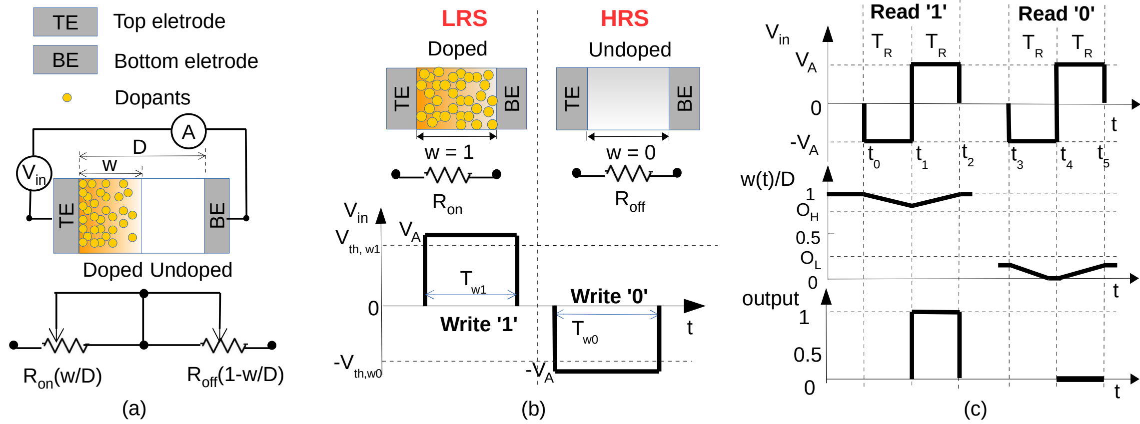

Fig. 3 The structure of a memristor cell and its write and read scheme. (a) memristor device structure and equivalent model [13]. (b) Write scheme. (c) Read scheme [14].

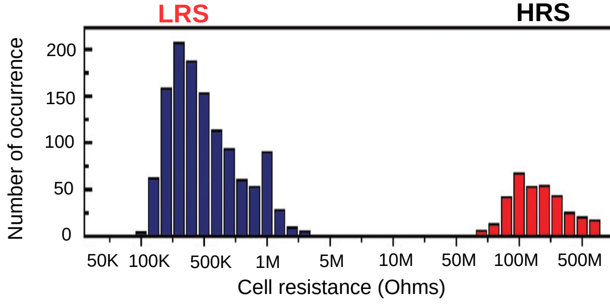

Fig. 4 Histogram of the HRS and LRS resistances variations extracted from a 40*40 nano-crossbar array (1600 devices) [16].

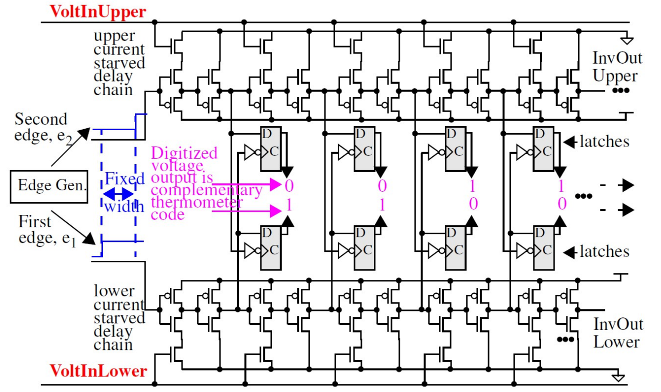

Fig. 5 Voltage-to-Digital Converter (VDC)

are used as the source of entropy. This is true since the read operations that take place during regeneration also change the resistance characteristics of the cells, which in turn increases the probability of a bit flip.

V. VDC and Proposed PUF Architectures

A. Voltage-to-Digital Converter (VDC)

The PUF structures that we propose require the measurement and digitization of a value proportional to the resistance of Memristor cells in the LRS. The proposed architectures, which are described in the following sections, provide a voltage from a voltage divider network(s) that is proportional to the LRS resistance.

The voltage-to-digital converter (VDC) shown in Fig. 5 is capable of digitizing these voltages [17]. The VDC has two voltage inputs, labeled VoltInUpper and VoltInLower, two digital inputs labeled e_1 and e_2, and two delay chains (upper and lower) connected to a sets of latches. The voltage inputs connect to NFET transistors inserted in series with the odd-numbered inverters of the delay chains. Voltages less than V_{DD} introduce additional delay through these inverters that is proportional to the applied voltage as an edge propagates down the inverter chains.

The function of the VDC is to create an 8-bit digital value between 0 and 128 that is related to the voltage present on the VoltInLower input. This voltage is derived from the voltage divider network and is always smaller than the supply voltage (V_{DD}). The digitization process is started by the Edge Generator, which launches a rising edge onto e1 and then after some delay, a second rising edge onto e2 as shown in the figure. Under the condition that the voltage on VoltInUpper is sufficiently larger than the voltage on VoltInLower, the e2 edge catches up and passes the e1 edge. The latches on the outputs of the even inverters in the delay chains record the point at which this happens as a thermometer code (TC). A TC is a sequence of '0's (or '1's) followed by a sequence of '1's (or '0's). The number of '1's (or '0's) in the TC reflects the magnitude of the difference between the two applied voltages. We refer to the number of '1's in the 128 latches connected to the lower chain as a TCV. In our proposed implementation, the voltage applied to VoltInUpper is V_{DD} as a means of ensuring that it is always larger than the voltage to be digitized on VoltInLower. The wide range of resistance variations that occur in the LRS states of Memristors cells produces a wide range of voltages that need to be digitized by the VDC. Moreover, TV environmental variations also impact the timing behavior of the VDC. The Edge Generator component of the VDC is used in a calibration process to ensure that the VDC is able to produce useful digital values under these conditions, where 'useful' is defined as values above 0 and less than the overflow value of 128. Calibration tunes the Δt between e1 and e2 edges, maximizing the sensitivity of the VDC to specific ranges of voltages, and allowing it to accommodate for TV variations. The transfer curve characteristics and calibration process are described below in the context of an example.

B. Memristor PUF

Fig. 6(a) shows the architecture proposed in [14] for a Memristor-based NVM, with the exception of the switch on the left side of the diagram (which is needed only for the PUF). The resistance of this switch, implemented as a pass gate with very wide transistors, is very low, e.g., on order of 50 Ohms or less.

This switch is closed when the memory is accessed for normal read and write operations. In this case, the Pulse Generator labeled V_{in} delivers pulses to a selected set (or word) of Memristor cells according to the diagrams shown earlier in Fig. 3. For normal read operations, the R/W Enable switch is set to the 'Read' position, which creates a voltage divider network between V_{in}, across the Memristor cell and resistor R_x to ground. The resistance of R_x is set to a value of approx. $(R_{off} + R_{on})/2$ so that V_X will be larger than V_{ref} (half of V_{in}) when the cell is programmed to its LRS and smaller than V_{ref} when programmed to HRS. In this way, V_O will be V_H (logic '1') when the Memristor cell is in LRS and V_L (logic '0') when the cell is in HRS. From the distributions shown in Fig. 4, the value of Rx would be approx. 10 MΩ.

The modifications shown in red in Fig. 6(b) are required in order to allow the Memristor memory to be used as a PUF. The large, low resistance switch is disabled and instead a high

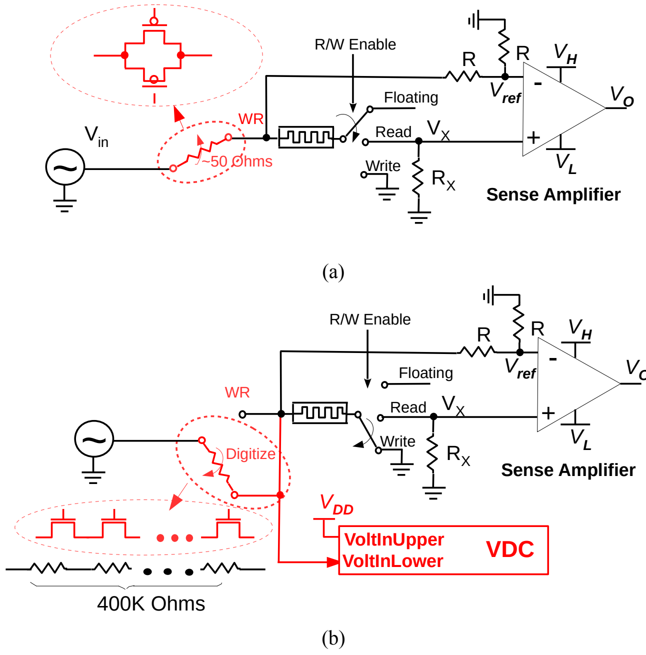

Fig. 6 (a) Circuit structure proposed Memristor memory [14]; (b) Modifications needed for proposed Memristor PUF.

resistance, approx. 400 KΩ, switch is enabled. This switch is also connected in series between the Pulse Generator and the Memristor array. The value of 400 KΩ is the resistance near the midpoint of the distribution of LRS programmed Memristor cells from Fig. 4. Therefore, when a Memristor cell that is programmed in its low resistance state is enabled, the voltage on the voltage divider network is a value between 200 mV and 882 mV (with V_{DD} at 1.0V). These values are obtained by using the extreme values of the LRS distribution in Fig. 4. For example, 200 mV is obtained from the voltage divider network expression (100 KΩ/500 KΩ). This voltage is delivered to the VoltInLower input of the VDC, as shown along the bottom of Fig. 6(b).

Most memory architectures are byte or word addressable, which means that multiple Memristor cells are accessed simultaneously. The arrangement shown in Fig. 6(b), on the other hand, assumes that each Memristor cell is individually addressable, i.e., the word-size of the PUF implementation is 1 bit. Therefore, an architecture level change is needed in addition to the components of Fig. 6(a) and (b) in order to convert the Memristor array into a PUF.

C. *Enrollment Algorithm*

As indicated earlier, the enrollment process leverages the random resistance variations in the Memristor cells as the source of entropy, and then uses the programmability of the Memristor cells to eliminate helper data. The enrollment algorithm that accomplishes these goals is given as follows:

1. The controller for the memory is instructed to program all Memristor cells to the low resistance state. This is accomplished as a 'normal' write 1 operation as described earlier with the large, low resistance pass gate switch enabled.
2. The controller is again instructed to sequence through a set of write operations but this time with the high resistance switch enabled and exactly one Memristor cell selected, i.e., the R/W enable signal is set to 'Write' while all other cells in the array are set to 'Floating'.
3. Immediately after the write pulse is asserted, a start signal is issued to the VDC to begin the digitization process.
4. The 8-bit digitized value from the VDC is stored in an on-chip SRAM memory at the address corresponding to the tested Memristor cell.
5. Once all cells in the Memristor array are digitized, a state machine creates a histogram of the digitized voltages stored in the SRAM. The histogram is created by using the digitized values as an address into a second on-chip SRAM, whose storage locations represent counters recording the number of instances of a particular digitized voltage.
6. A state machine parses the histogram data from low to high address, adding up the counter values. The memory address of the median value, which partitions the array of elements into two equal-sized groups, is recorded.
7. The state machine then parses the first SRAM, comparing the stored digitized voltage with the median. The Memristor array is again placed in normal write mode and those cells whose value exceeds the median are re-programmed to the HRS.
8. A bistring is constructed using a sequence of normal read operations, which are designed to preserve the LRS or HRS of the programmed Memristor cells. The sequence of read addresses can be generated as a linear sequence or by using a linear-feedback-shift-register to generate the sequence pseudo-randomly.

The ordering of the Memristor cells from left to right within the histogram is random for each chip, and therefore, the bitstrings will be unique across chips. Also, the large threshold between the two distributions makes it possible for the bit generation algorithm to succeed in reliably making the same decision about whether the Memristor cell is in a LRS or HRS, thereby eliminating the need for helper data.

Note that resistor divider network reduces the 'write' voltage during the digitize operation, in most cases to a value below the threshold shown in Fig. 3(b). Therefore, changes in the actual resistance value are likely to be very small. However, the enrollment process as described is robust to these types of resistance changes so they are of no consequence.

VI. Evaluation Using Measured Data

This section demonstrates the practicability of the enrollment process using the measured data from [16]. Fig. 7 shows that resistance variations of the LRS programmed Memristor devices ranging from approx. 100 KΩ to 3 MΩ, and

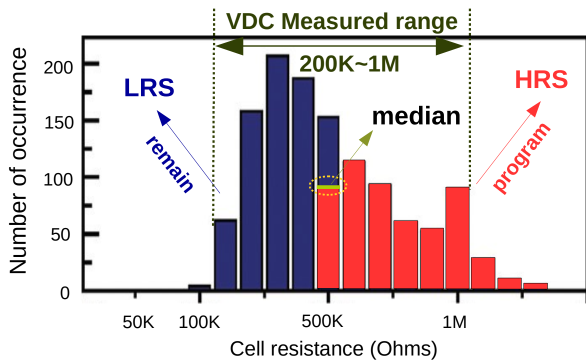

Fig. 7 VDC measured range for the measured LRS data profile from 1220 memristor cells in [16].

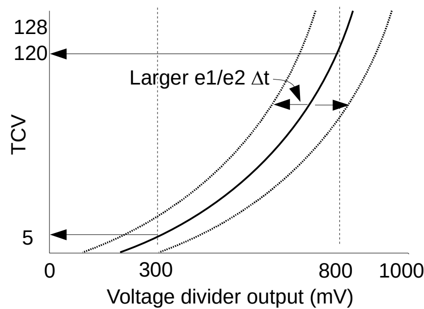

Fig. 8 Typical transfer Curves for VDC.

the profile does not need to be Gaussian. A robust feature of our proposed median finding algorithm is that we do not need to build the voltage divider network and VDC to digitize this entire range. In fact, only the values in the middle of the distribution, i.e., in the range of 200 KΩ to 1 MΩ, need to produce non-underflow (0) and non-overflow (128) TCVs within the VDC.

The transfer curves in Fig. 8 indicate that the VDC operates best for VoltInLower values in the range of 300 mV to 800 mV (for V_{DD} of 1V), where it produces TCVs in the range from 5 to 120. Note that this range can be adjusted using calibration to accommodate process and TV variations, as shown by the dotted curves. Calibration tunes the Δt between e1 and e2, effectively shifting the curves horizontally. Setting the high resistance switch in Fig. 6(b) to approx. 400 KΩ produces voltages of 333 mV when the Memristor cell is 200 KΩ and 714 mV for Memristor cells at 1 MΩ. This range of 333 mV to 714 mV fits within the digitization range of 300 mV to 800 mV, and therefore all cells in this example produce non-underflow and non-overflow digital values.

VII. Conclusion

In this paper, we propose a Memristor-based PUF design that is capable of eliminating helper data. A special enrollment process is used to digitize and characterize the cell resistance distribution as a means of partitioning the array of elements into two components. The elements belonging to the lower component are programmed to their LRS while those in the upper component are programmed to their HRS. The large margin that typically exists between the LRS and HRS distributions enables the bitstring to be reliably regenerated without the need for publicly stored helper data. This feature of the NVM-PUF makes it attractive from a practical perspective because it eliminates the overhead and security issues associated with helper data.

VIII. Acknowledgments

This research is supported in part by NSF grant CNS-1018748, CNS-1054744 and DUE-1245756.

IX. References

[1] B. Gassend, et al., "Controlled Physical Random Functions," *Conference on Computer Security Applications*, 2002.
[2] B. Skoric, et al., "Robust Key Extraction from Physical Uncloneable Functions", Chapter in *Applied Cryptography and Network Security*, 2005.
[3] R. Maes, et al., "Low-overhead Implementation of a Soft Decision Helper Data Algorithm for SRAM PUFs," *CHES*, 2009, pp. 332–347.
[4] Z. Paral, et al., "Reliable and Efficient PUF-based Key Generation using Pattern Matching," *HOST*, 2011, pp. 128-133, Jun. 2011.
[5] J. Delvaux. et al., "Attacking PUF-Based Pattern Matching Key Generators via Helper Data Manipulation", Cryptology ePrint Archive: Report 2013/566
[6] R. Chakraborty, et al., "A Transmission Gate Physical Unclonable Function and On-Chip Voltage-to-Digital Conversion Technique", in Proceedings of the 50th Design Automation Conference, *DAC* '13, 2013, May. 2013.
[7] P. Koeberl, et al., "Memristor PUFs: A New Generation of Memory-based Physically Unclonable Functions", *DATE*, 2013, pp. 428-431, Mar. 2013.
[8] G. S. Rose, et al., "Hardware Security Strategies Exploiting Nanoelectronic Circuits", *ASP-DAC*, 2013, pp. 368-372, Jan. 2013.
[9] G. S. Rose, et al., "Foundations of Memristor Based PUF Architectures," *NANOARCH*, July 2013.
[10] G. S. Rose, et al., "A Write-Time Based Memristive PUF for Hardware Security Applications", *ICCAD*, 2013, pp. 830-833, Nov. 2013.
[11] O. Kavehei, et al., "mrPUF: A Memristive Device based Physical Unclonable Function", *CoRR*, 2013.
[12] J. J. Yang, et al., "Memristive Devices for Computing," *Nature Nanotechnology.*, vol. 8, pp. 13–24, Jan. 2013.
[13] D. B. Strukov et al, "The Missing Memristor Found", in *Nature*, volume 453, pages 80–83, 2008.
[14] Y. Ho, et al., "Dynamical Properties and Design Analysis for Nonvolatile Memristor Memories", Trans. *CAS*, 58-I(4):724–736, 2011.
[15] Q. Xia, et.al. "Memristor-CMOS Hybrid Integrated Circuits for Reconfigurable Logic", *Nano Letters*, vol. 9, no. 10, 2009.
[16] K. Kim, et al., "A Functional Hybrid Memristor Crossbar-array/CMOS System for Data Storage and Neuromorphic Applications," *Nano Letters*, vol. 12, no. 1, pp. 389–395, 2011.
[17] J. Ju, et al., "Stability Analysis of a Physical Unclonable Function based on Metal Resistance Variations", *HOST*, 2013, pp. 143-150.