Leakage-resilient memory-based physical unclonable function using phase change material

Zhang, Le; Chang, Chip-Hong; Cabrini, Alessandro; Torelli, Guido; Kong, Zhi Hui

2014

Zhang, L., Chang, C.-H., Cabrini, A., Torelli, G., & Kong, Z. H. (2014). Leakage-resilient memory-based physical unclonable function using phase change material. 2014 International Carnahan Conference on Security Technology (ICCST), 1-6.

<https://hdl.handle.net/10356/105042>

<https://doi.org/10.1109/CCST.2014.6987047>

© 2014 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at: [<http://dx.doi.org/10.1109/CCST.2014.6987047>].

Downloaded on 26 Apr 2023 23:29:22 SGT

Leakage-Resilient Memory-based Physical Unclonable Function Using Phase Change Material

Le Zhang*, Chip-Hong Chang*, Alessandro Cabrini[†], Guido Torelli[†]
and Zhi Hui Kong*

*School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore

[†] Dipartimento di Ingegneria Industriale e dell'Informazione, University of Pavia, Pavia, Italy

Abstract—Memory-based Physical Unclonable Function (MemPUF) emerged as a replacement for traditional key preservation primitives to overcome the susceptibility of secret keys to physical attacks. Recent experiments demonstrated that even some MemPUFs can be physically attacked by exploiting their side-channel information. In this paper, we formulate an adversary model for a prediction attack that takes advantage of the side-channel information leaked from a MemPUF. Based on this pivotal insight, we propose countermeasures to enhance the resilience of MemPUFs against such a kind of attack, and introduce a security-enhanced MemPUF design using phase change material. Our analysis demonstrated the effectiveness of our proposed scheme against the measurement-prediction attack given an adversary with certain bounded attack capability.

Index Terms—Hardware Security, Physical Unclonable Function, Phase Change Memory, Leakage Resilience Cryptography

I. INTRODUCTION

Memory-based Physical Unclonable Function (MemPUF) has been entrusted as robust device fingerprint and hardware random key generator for security applications. Recent research demonstrated that some MemPUFs can be physically attacked. For instance, Yossef *et al.* [1] reported that SRAM-based memory is clonable if the data remanence decay side-channel can be used for a fault-injection attack. There are also some attacks on other types of Physical Unclonable Functions (PUFs) that exploit side-channel information to obtain the secrets preserved in a PUF device [1]–[4]. These recent findings insinuate that it is over-optimistic to consider MemPUF as a foolproof black-box to evaluate the security of a MemPUF or a crypto-system based on a MemPUF. More realistically, a MemPUF should be regarded as a physical device that may leak measurable side-channels to the outside world [5]. In addition, in security analysis, it is necessary to take into consideration a high-level adversary who is capable of obtaining certain information leaked from the MemPUF.

In this paper, we model a high-level adversary who is able to physically measure and then predict the response bits of a MemPUF. Some approaches are also suggested as countermeasures to enhance the security of MemPUF against this kind of attack. It is costly to rely completely on conventional methods such as protective coating layers, error detection codes, etc. to protect MemPUFs from physical attacks. Instead, we circumvent the problem by limiting the capability of the adversary and refreshing the response bits

that can be generated from a MemPUF by designing the MemPUF with alternative materials, so that the probability of a successful prediction attack is correspondingly lowered. Our contributions are summarized as follows.

- We formulate a high-level adversary model that can physically measure a MemPUF and then perform a prediction attack on the output responses of the MemPUF.
- We present a novel MemPUF design using Phase Change Memory for enhanced leakage resilience.
- We evaluate our proposed MemPUF and analyze its security against the measurement-prediction attack.

The rest of the paper is organized as follows. Section II introduces the fundamentals of PUF and MemPUF. Section III formulates the adversary model and the measurement-prediction attack performed by the adversary. Strategies for leakage-resilient enhancement are suggested in Section IV, and the proposed PCM-based MemPUF and its security analysis are presented in Section V. The paper is concluded in Section VI.

II. MEMPUF FUNDAMENTALS

A. Physical Unclonable Function

A PUF (denoted as $\mathcal{M}$) is a probabilistic function, which takes advantage of the intrinsic physical disordered properties of a device resulting from fabrication process uncertainties to map the input challenge C to the output response R . Owing to the impacts of noise, its response to an identical challenge may not be stable over time. A common practice to improve the stability and randomness of PUF responses is to post-process the raw PUF responses by using a *fuzzy extractor*. A PUF works in two phases with the incorporation of a fuzzy extractor [6] as follows.

- *Enrollment phase* - when the PUF is used for the first time, its raw response R is XOR-ed with a helper data (H_1), and the result is encoded as a code word W by the Error-Correction Code (ECC) in the fuzzy extractor. The code word W is then processed by a universal hash function specified by another helper data (H_2) to produce the final key K .
- *Regeneration phase* - the raw response R' is re-generated from $\mathcal{M}$. Due to noise effects, $R' \neq R$. R' is then XOR-ed with H_1 and decoded by the ECC to produce W . W can be correctly reproduced only if $R \oplus R' \leq t$ where t

is the random error correctable by the ECC. W is then processed by the hash function specified by H_2 to output the final key K .

B. Memory-based PUF

MemPUF is a kind of PUF that is constructed based on a memory or a memory-like array [7], [8]. The challenge of a MemPUF is the address (c_i) to access a memory cell, and the response bit is the bit (r_i) read from the memory cell selected by c_i . A response string R of a MemPUF can therefore be formed by concatenating the response bits generated from the memory cells accessed by a challenge set, i.e., $R = r_1 || r_2 || \dots || r_l$.

III. MEASUREMENT-PREDICTION ATTACK ON MEMPUF

A. Key generation scheme

First of all, we assume that the MemPUF $\mathcal{M}$ is used for a key generation scheme by an authority ($\mathcal{U}$) in certain security applications. Specifically, the scheme consists of the three following functions.

- $\text{KeyGen}[C, (H_1, H_2), t] \rightarrow K$: this function generates the secret key K from $\mathcal{M}$ given the challenge and the specifications of the fuzzy extractor module. This function consists of two sub-functions:
 - $R \leftarrow \text{RespGen}(C)$, which produces the response R from $\mathcal{M}$ given the challenge C ;
 - $K \leftarrow \text{FuzExt}(R, H_1, H_2, t)$, which outputs the key K by post-processing R assisted by the helper data H_1 and H_2 ; the error correction capability of the ECC in the fuzzy extractor is specified by t .
- $\text{KeySync}(K, \mu_{K'}) \rightarrow \mu_K$: KeySync is the function to synchronize the information about K (denoted as μ_K) with $\mathcal{U}$. In the conventional PUF system described in Section II-A, KeySync may refer to the enrollment phase in which K as well as the helper data H_1 and H_2 are sent to $\mathcal{U}$ as μ for future verification.
- $\text{KeyVerify}(K, \tilde{K}, \mu_K) \rightarrow \{0, 1\}$: KeyVerify checks the validity of the key before it is used. Its output is 1 if the verification is successful, i.e., if the key aligns with the information μ_k , whereas it is 0 otherwise.

B. Adversary model

We assume an adversary $\mathcal{A}$ who is able to perform certain physical attacks on $\mathcal{M}$ in order to forge the device and cheat $\mathcal{U}$. $\mathcal{A}$ is assumed to be able to physically possess a MemPUF ($\mathcal{M}$) for a period of time t_m after it has been enrolled. We assume that, in the key generation scheme, a set of MemPUF cells, $\mathcal{M}_m^l$, are selected by m challenge sets (each containing l challenges) to generate m l -bit raw response strings in the enrollment phase, and the enrollment is perfectly secure, i.e., $\mathcal{A}$ has no idea about the content of $\mathcal{M}_m^l$ before he/she measures it. The i -th response string, where $1 \leq i \leq m$, is denoted as R_i^l . During t_m , $\mathcal{A}$ can leverage on any possible side-channel information leaked from $\mathcal{M}$ to perform measurements on the MemPUF cells. $\mathcal{A}$ is allowed to explore the whole space of $\mathcal{M}$ within t_m , and on average

Fig. 1. Measurement-prediction attack performed by the adversary $\mathcal{A}$.

η MemPUF cells can be measured within a unit time. The number of cells (represented by λ) that $\mathcal{A}$ can measure from $\mathcal{M}$ during t_m is therefore equal to $t_m \times \eta$. The set of cells measured by $\mathcal{A}$ is denoted as $\mathcal{M}^\lambda$. Due to the inaccuracies in measurements, the response bits measured by $\mathcal{A}$ may not be exactly the same as those generated from $\mathcal{M}$. We denote the probability that the response bit measured is the same as that generated from a cell by α (which we will refer to as *measurement accuracy* hereafter). The adversary in this model is fully specified by the tuple (t_m, η, α) .

C. Measurement prediction Attack

After $\mathcal{A}$ has performed measurements on $\mathcal{M}$ and collected the data from $\mathcal{M}^\lambda$, $\mathcal{A}$ will predict the response generated from $\mathcal{M}$, R , and then calculate the corresponding secret key, K , with the assistance of H_1 and H_2 . With the predicted keys, $\mathcal{A}$ tries to forge the device to deceive or impersonate $\mathcal{U}$. The behavior of $\mathcal{A}$ when measuring $\mathcal{M}^\lambda$ and predicting response bits is called *measurement-prediction (MP) attack* (see Fig. 1). More specifically, the attack is called

- *partial MP attack*, if $\lambda < |\mathcal{M}|$;
- *full MP attack*, if $\lambda = |\mathcal{M}|$.

Apparently, if the response predicted by $\mathcal{A}$ (denoted as $\tilde{R}_i^l$) matches the response, R_i^l , generated from $\mathcal{M}$, the corresponding secret keys will become known to $\mathcal{A}$. In this case, if $\mathcal{A}$ sends the predicted key, $\tilde{K}$, to $\mathcal{U}$ for any application, the function of $\text{KeyVerify}(K, \tilde{K}, \mu_K)$ will output 1. The MP attack is said to be *successful* if $\text{KeyVerify}(K, \tilde{K}, \mu_K) = 1$; Otherwise, it *fails*.

Now we analyze the security of a MemPUF system against an MP attack performed by a (t_m, η, α) -adversary. Assuming the cells selected for response generation belong to $\mathcal{M}^\lambda$, $\mathcal{A}$ can predict the response bits by directly using the measurement results. Due to the inaccuracy of measurements, $\tilde{R}_i^l \neq R_i^l$. $\mathcal{A}$ can reconcile the noisy response $\tilde{R}_i^l$ to R_i^l by means of H_1 if the error bits induced by the measurement inaccuracy are within the correction capability t of the ECC incorporated in the fuzzy extractor. On the other hand, if the cells selected by $\mathcal{U}$ for key regeneration do not belong to $\mathcal{M}^\lambda$, $\mathcal{A}$ needs to guess the response string R_i^l . The probability of correctly guessing the response bit generated from a cell is at least $2^{-\beta}$, where β is the entropy of the response bit in a MemPUF cell.

Therefore, for a partial MP attack, suppose l cells are selected by $\mathcal{U}$ from $\mathcal{M}_m^l$ for key generation and q out of l cells can be found in $\mathcal{M}^\lambda$, the probability of a successful

partial MP attack (denoted as $\mathbf{Pr}_{partial}^{succ}$) can be calculated as

$$\mathbf{Pr}_{partial}^{succ} = \frac{\binom{\lambda}{q} \binom{|\mathcal{M}|-\lambda}{l-q}}{\binom{|\mathcal{M}|}{l}} \sum_{i=0}^t (1-\alpha)^i \alpha^{q-i} \times (2^{-\beta})^{l-q} \quad (1)$$

In the worst-case, $q = \lambda$ if $l > \lambda$, and $q = l$ if $l \leq \lambda$, i.e., $\mathbf{Pr}_{partial}^{succ}$ is equal to its maximum value when all of the l cells selected for response generation belong to $\mathcal{M}^\lambda$.

For a full MP attack, $\mathcal{A}$ can predict any of the responses generated from the cells in $\mathcal{M}_m^l$ since he/she has measured all the response bits of these cells. In this case, the probability of a success ($\mathbf{Pr}_{full}^{succ}$) is

$$\mathbf{Pr}_{full}^{succ} = \sum_{i=0}^t (1-\alpha)^i \alpha^{l-i} \quad (2)$$

IV. LEAKAGE-RESILIENCE ENHANCEMENT

A. Making a full MP attack difficult

Generally, given that $\mathcal{A}$ has the capability of measuring $\mathcal{M}$ with no limitation, completely preventing a full MP attack is non-trivial unless the memory array has infinite size. However, measures can be taken to make a full MP attack *difficult*:

- 1) limiting the capability of $\mathcal{A}$ for physical measurements, i.e., reducing η ;
- 2) reducing the time window (t_m) during which $\mathcal{A}$ is allowed to perform his/her measurement on $\mathcal{M}$.

η is primarily dependent on the technology used by $\mathcal{A}$ for the measurement and the complexity of the MemPUF. Other conditions being equal, the more complicated and delicate the MemPUF is, the more difficult it is to measure a cell within a given time. Increasing the complexity of a MemPUF can be achieved by using more advanced fabrication technology, i.e., by shrinking the feature size of the fabrication technology and/or using emerging materials that has not (yet) been fully investigated. On the other hand, reducing t_m may be achieved by incorporating a detecting mechanism that periodically examines if there are attempts to tamper with the MemPUF.

It is worth pointing out that reducing either η or t_m cannot prevent a full-measurement prediction attack if $\mathcal{A}$ is allowed to measure $\mathcal{M}$ for an unlimited number of times (each time lasts at most t_m). This is because $\mathcal{A}$ will eventually measure the whole MemPUF array no matter how small η and t_m are. After the whole response space is exhausted by $\mathcal{A}$, the MemPUF will be compromised.

B. Making a partial MP attack difficult

Assuming $\mathcal{A}$ is able to physically hold the MemPUF only for a limited amount of time, he/she may just be able to achieve a partial MP attack if η and t_m are small. Also, increasing the entropy per cell (i.e., β) helps to reduce $\mathbf{Pr}_{partial}^{succ}$ according to Eq. (1). To increase the entropy contained in each cell, other possible physical properties of MemPUF devices in addition to the intrinsic process variations can be exploited.

C. Physically reconfigurable MemPUF with multi-bit response per cell and slothful response generation

In this Subsection, we introduce a MemPUF that has enhanced leakage resilience against MP attacks performed by a (t_m, η, α) -adversary. Our proposed MemPUF achieves this goal by means of *physical reconfigurability*, *multiple response bit capacity per cell*, and *slothfulness of response bit generation*.

- **Physical reconfigurability:** by making use of certain effects, the physical properties of the MemPUF device may be altered randomly so as to alter the challenge-response mapping characteristic of the MemPUF. This method is called *physical reconfiguration* because it does not rely on any additional hardware primitive such as hash function or block cipher module [9]. There is another category of reconfiguration method, namely the “logical reconfiguration”, which does not help to enhance leakage resilience. This is because after $\mathcal{A}$ has measured the raw responses from $\mathcal{M}$, he/she is able to obtain any form of logically reconfigured response bits.
- **Multiple response bit capacity per cell:** multiple response bits can be generated from a single MemPUF cell and, thus, the entropy contained in the cell will be increased.
- **Slothfulness of response bit generation:** unlike conventional MemPUFs that produce response bits in nanoseconds, the proposed MemPUF will take longer time to output one bit from a cell. This in turn reduces the value of parameter η of $\mathcal{A}$, thus making an MP attack more difficult to be accomplished.

To be compatible with physical reconfigurability, a new basic function is introduced in the key generation scheme, that is, $\text{KeyReconf}(S, \mathcal{M}) \rightarrow \mathcal{M}_S$, where the *configuration state* S refers to the digitized physical effects used to reconfigure the MemPUF $\mathcal{M}$ [9], [10]. $\mathcal{M}_S$ correspondingly refers to the MemPUF that is reconfigured by the physical effects determined by S . $\mathcal{U}$ will be updated about the key and the configuration state by the synchronization function KeySync .

V. PHASE CHANGE MEMORY BASED MEMPUF

There exist several implementations of physically reconfigurable PUFs [10], [11], multi-response-bit PUFs [11], and PUFs with slothful response bit generation [12]. However, no single PUF integrating the three features based on the above-mentioned insights has been presented so far in the literature. In this Section, we will introduce a “3-in-1” MemPUF based on the emerging Phase Change Memory (PCM) technology.

A. PCM basics and mechanism of PCM-based MemPUF

PCM is considered as one of the most promising technologies for next-generation memories. When compared to conventional CMOS memories such as SRAM, PCM has higher integration density and better scalability [13]. PCM utilizes a chalcogenide alloy, typically GeSbTe (GST) as storage element [Fig. 2(a)]. The state of GST can be switched to either an *amorphous* (high resistivity) or a *crystalline* (low resistivity) phase. The process of amorphization (RESET) and

Fig. 2. PCM basics: (a) structure of the PCM array and the *GST*-based storage element; (b) electrical pulse regulator for read/write operation on PCM cells; (c) temperature characteristics for RESET and SET programming operations.

Fig. 3. Programming dynamics of a SET operation on a PCM cell initialized to a consistent RESET state: (a) PCM resistances distribute randomly when an identical electrical pulse is applied owing to process variations; (b) the programming result of a certain PCM cell varies according to the characteristics of the electrical pulse.

crystallization (SET) of the GST alloy can be induced by applying adequate (different) programming voltages to the PCM cell as shown in Fig. 2(b). For RESET operation, a pulse with short duration, high amplitude, and short fall time (fast quench) is applied to produce a high enough temperature (thanks to a heater) to melt and amorphize the GST layer. For SET operation, a pulse with long duration and intermediate amplitude is used to mildly increase the temperature and sufficiently crystallize the GST layer, or a high magnitude and long fall time pulse is applied to melt and, then, crystallize the GST material [Fig. 2(c)] [13].

Scaling down the technology to fabricate PCM devices leads to an aggravation of process induced variations. Variations of PCM device parameters cause variations in the PCM dynamics. As can be seen in Fig. 3, when identical programming pulses are applied, the programmed resistance values of PCM cells spread stochastically. In addition, changing the programming pulse amplitude and/or duration leads to different programming results for the same PCM cell (Fig. 3). Suppose a selected PCM cell is initialized to a certain resistance value (denoted as Res_{start}) and is then iteratively programmed until its resistance reaches a pre-determined target value (denoted as Res_{target}), as shown in Fig. 4: the number of programming pulses needed to make the resistance value converge to Res_{target} will be randomly distributed [14]. The number of programming pulses used can therefore

Fig. 4. Response generation of the proposed PCM-based MemPUF.

Fig. 5. Response generation by using the proposed PCM-based MemPUF in enrollment and regeneration phases.

be encoded as response bits generated from the PCM cell. Assuming the MemPUF is synchronized by an internal timer $\sigma \in [0, \infty)$ so as to allow the enrollment cycle to be repeated at given time intervals, the process of response generation by using the proposed PCM-based MemPUF is illustrated in Fig. 5.

Since the response generated from a PCM cell is not merely dependent on fabrication process variations, but also on Res_{start} , Res_{target} , and the characteristics of the pulses that are used for the iterative programming process, by changing these parameters, the response generated from an identical PCM cell will be different. Moreover, the response bits generated from the PCM-based MemPUF can be periodically refreshed so that predicting any of the new key from the old ones will be difficult. After a time interval equal to a

Fig. 6. Adversary model when considering periodical key refreshment of the MemPUF.

predetermined value Γ , the MemPUF is enrolled again, i.e., new values of Res_{target} , $V_{p,(1)}$ and $t_{p,(1)}$ (Fig. 4) are produced from a random source¹. The responses generated from the cells of $\mathcal{M}_m^l$ are then updated accordingly. To achieve a slothful response generation from the PCM based MemPUF, SET programming is selected since it takes longer time when compared to its RESET counterpart. Also, considering the Johnson-Mehl-Avrami-Kolmogorov equation [15] and the temperature dependence of the coefficients in this equation according to Arrhenius law, the starting programming voltage $V_{p,(1)}^{\Gamma(j)}$, as well as the voltage step ΔV , can be made sufficiently small so that the time for making the resistance of a cell converge to the target value is prolonged. The output response, which is the number of pulses used for making the cell converge to a predetermined value, can be decoded as a multi-bit output so as to increase its entropy [9].

B. Security analysis

Since our proposed MemPUF can be refreshed periodically, we assume the adversary $\mathcal{A}$ can perform the measurement at any time between two consecutive enrollment operations and predict the output responses within (labeled as “prediction-1” in Fig. 6) or outside the enrollment cycle (labeled as “prediction-2” in Fig. 6).

1) *Resistance to full MP attack*: The proposed MemPUF makes a successful full MP attack difficult by means of the following actions.

- *Reducing η* - Compared to conventional MemPUFs, the proposed MemPUF can be fabricated by using more scaled technology [16]. The difficulty in measuring a chip with smaller feature size is higher, which may make η smaller if the same measurement accuracy must be guaranteed. Besides, the programming process for response generation is iterative, and each iteration takes a time interval $t_{p,(i)}^{\Gamma(j)}$. Therefore, the adversary $\mathcal{A}$ should take at least as much time as $1/\eta = t_{itr} + t_{del}$ to measure a cell, where $t_{itr} = \sum_{i=1}^{\omega} t_{p,(i)}^{\Gamma(j)}$ is the time taken by the iterative programming process and $t_{del} \geq 0$ is the additional time (if any) taken by $\mathcal{A}$ to complete the measurement.
- *Limiting t_m* - The responses generated from the MemPUF are refreshed at a time interval equal to Γ and subse-

¹The random source may be an on-chip random number generator that controls a digital-to-analog converter to produce the resistance, voltage, and time duration values.

Fig. 7. $\mathbf{Pr}^{succ}$ for different values of Γ and t_{del} ($l = 64$).

quently re-enrolled. Measuring the MemPUF when it is being re-enrolled is assumed to be prohibited. Therefore, even if $\mathcal{A}$ succeeds in a full MP attack before the j -th re-enrollment, the correctly predicted challenge-response pairs expire after Γ and cannot be used to infer the response bits generated from the MemPUF after the re-enrollment. In other words, t_m is limited to Γ .

2) *Resistance to partial MP attack*: If $\mathcal{A}$ cannot succeed in a full MP attack, the probability of a successful partial MP attack can be reduced by increasing the entropy contained in each cell. In our proposed PCM-based MemPUF, the output response is produced by encoding the number of iterations needed to make a PCM cell converge to the target resistance value. The bit-length of the response generated is determined by the quantization resolution of the encoder. For example, if the possible range of ω is $[10, 20]$, one bit can be produced from the i -th cell with the rule

$$\begin{cases} r(i) = 1, & \text{if } \omega \geq 15 \\ r(i) = 0, & \text{if } \omega < 15 \end{cases} \quad (3)$$

Two bits can be produced with the following rule:

$$\begin{cases} r(i) = 11, & \text{if } \omega \geq 18 \\ r(i) = 10, & \text{if } 15 \leq \omega < 18 \\ r(i) = 01, & \text{if } 12 \leq \omega < 15 \\ r(i) = 00, & \text{if } 10 \leq \omega < 12 \end{cases} \quad (4)$$

It should be noted that increasing the quantization resolution excessively will degrade the reliability of the MemPUF due to the ensuing reduction of noise margin.

C. Evaluation of results and discussion

Let β_s and β_d denote the entropy per cell before and after the cell is reconfigured, respectively (these parameters may be called *static* and *dynamic* entropy, according to the corresponding definitions of predictability [9]). The probability $\mathbf{Pr}^{succ}$ of successful prediction for “prediction-1” and “prediction-2” as illustrated in Fig. 6 can be calculated as follows:

- “prediction-1” -
- $$- \mathbf{Pr}^{succ} = \frac{\binom{\lambda}{q} \binom{|\mathcal{M}| - \lambda}{l - q}}{\binom{|\mathcal{M}|}{l}} \sum_{i=0}^t (1 - \alpha)^i \alpha^{q-i} \times (2^{-\beta_s})^{l-q}$$
- for a *partial MP*;

Fig. 8. $\Pr^{succ}$ for different values of l (64, 128, and 256) given a fixed Γ .

$$- \Pr^{succ} = \sum_{i=0}^t (1 - \alpha)^i \alpha^{l\beta_s - i} \text{ for a full MP.}$$

$$\bullet \text{ “prediction-2” } - \Pr^{succ} = (2^{-\beta_d})^l.$$

With appropriate values of Γ , t_p , and quantization resolution of the pulse count for response bit generation, the probability of success of a full MP attack on the proposed PCM-based MemPUF will be very low.

To demonstrate the effectiveness of the proposed approach, we now evaluate the security of our proposed scheme against a (t_m, η, α) -adversary. The parameters of the adversary are constrained by the PCM MemPUF system as:

- 1) $1/\eta = \sum_{i=1}^{\omega} t_{p,(i)} + t_{del}$;
- 2) $\alpha \leq 1 - \text{BER}$;
- 3) $t_m \leq \Gamma$.

where BER refers to the Bit-Error Rate of the generated response string owing to noise effects.

We evaluate $\Pr^{succ}$ with the assumptions of some qualities of the PCM based MemPUF based on the test results obtained from a 180-nm PCM experimental chip [9], [17]. We use an array consisting of 1024 cells for the MemPUF, i.e., $|\mathcal{M}| = 1024$. The resolution used for quantizing the programming pulse count is 3-bit. The worst-case measurement accuracy is given by $\max(\alpha) = 1 - \min(\text{BER})$. A $\min(\text{BER})$ of 0.15 is assumed, which leads to $\max(\alpha) = 0.85$. The values of β_s and β_d are assumed to be 2.5 and 2.0, respectively². In the worst case, all of the l cells addressed to regenerate responses belong to $\mathcal{M}^\lambda$ (see Section III-C) and the number of selected cells q is equal to λ or l (depending on the relationship between l and λ). t_p is assumed to be 1 ms, and 50 iterations are assumed to make a cell converge to the target resistance. t_{del} is assumed to range from $1\mu s$ to $1s$. Γ is made to vary from several minutes to several days to analyze its impact over $\Pr^{succ}$.

The results for generating response string using 64 cells selected from the MemPUF array are shown in Fig. 7. It can be seen that reducing either Γ or $1/\eta$ will make the full MP attack more difficult because more efforts are needed by $\mathcal{A}$ to perform a prediction attack. For example, by setting $\Gamma = 1$ hour, the

MemPUF is robust against an adversary with a measurement capability $\eta \approx 0.7$ cell/s. To further enhance the resilience, one can increase l for response generation so that $\Pr^{succ}$ for a partial MP attack becomes even lower. Fig. 8 depicts $\Pr^{succ}$ by setting Γ to 7 days for different values of l .

VI. CONCLUSION

In this paper, we formulate an adversary model based on the tenable full and partial MP attacks leveraging side-channel information leaked from MemPUF. As a countermeasure, we propose a novel MemPUF based on PCM. Our method limits the rate of measuring a MemPUF cell, thus making an MP attack difficult. Moreover, the MemPUF is periodically refreshed so that the probability of correctly predicting the response is lowered and, in the case of a successful MP attack, the usefulness of the predicted responses expires shortly after they are successfully predicted. Since the refreshment relies totally on the physical parameters of MemPUF, there is no efficient way for the adversary to predict the response bits other than repeating physical measurements.

REFERENCES

- [1] Y. Oren *et al.*, “On the effectiveness of the remanence decay side-channel to clone memory-based PUFs,” in *CHES*, 2013, pp. 107–125.
- [2] J. A. Halderman *et al.*, “Lest we remember: cold-boot attacks on encryption keys,” *Commun. ACM*, vol. 52, no. 5, pp. 91–98, 2009.
- [3] D. Merli *et al.*, “Localized electromagnetic analysis of RO PUFs,” in *IEEE HOST*, 2013, pp. 19–24.
- [4] C. Helfmeier, C. Boit, D. Nedospasov, and J.-P. Seifert, “Cloning physically unclonable functions,” in *IEEE HOST*, 2013, pp. 1–6.
- [5] F. Armknecht *et al.*, “Memory leakage-resilient encryption based on physically unclonable functions,” in *Towards Hardware-Intrinsic Security*. Springer, 2010, pp. 135–164.
- [6] Y. Dodis *et al.*, “Fuzzy extractors: How to generate strong keys from biometrics and other noisy data,” in *EuroCrypt*, May 2004, pp. 523–540.
- [7] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, “FPGA intrinsic PUFs and their use for IP protection,” in *CHES*, Sept. 2007, pp. 63–80.
- [8] D. Holcomb *et al.*, “Power-up SRAM state as an identifying fingerprint and source of true random numbers,” *IEEE Trans. Comput.*, vol. 58, no. 9, pp. 1198–1210, Sept. 2009.
- [9] L. Zhang, Z. H. Kong, C.-H. Chang, A. Cabrini, and G. Torelli, “Exploiting process variations and programming sensitivity of phase change memory for reconfigurable physical unclonable functions,” *IEEE Trans. Inf. Forens. Security*, vol. 9, no. 6, pp. 921–932, June 2014.
- [10] K. Kursawe, A. Sadeghi, D. Schellekens, B. Skoric, and P. Tuyls, “Reconfigurable physical unclonable functions-enabling technology for tamper-resistant storage,” in *IEEE HOST*, Jul. 2009, pp. 22–29.
- [11] L. Zhang *et al.*, “PCKGen: a phase change memory based cryptographic key generator,” in *IEEE ISCAS*, May 2013, pp. 1444–1447.
- [12] U. Ruhrmair *et al.*, “Applications of high-capacity crossbar memories in cryptography,” *IEEE Trans. Nanotechnol.*, vol. 10, no. 3, pp. 489–498, 2011.
- [13] H. P. Wong *et al.*, “Phase change memory,” *Proc. IEEE*, vol. 98, no. 12, pp. 2201–2227, Dec 2010.
- [14] S. Braga *et al.*, “Voltage-driven partial-reset multilevel programming in phase-change memories,” *IEEE Trans. Electron Devices*, vol. 57, no. 10, pp. 2556–2563, Oct. 2010.
- [15] M. Avrami, “Granulation, phase change, and microstructure kinetics of phase change. III,” *J. Chemical Physics*, vol. 9, no. 2, pp. 177–184, 1941.
- [16] U. Russo *et al.*, “Modeling of programming and read performance in phase-change memoriespart I: Cell optimization and scaling,” *IEEE Trans. Electron Devices*, vol. 55, no. 2, pp. 506–514, 2008.
- [17] F. Bedeschi *et al.*, “4-Mb MOSFET-selected μ trench phase-change memory experimental chip,” *IEEE J. Solid-State Circuits*, vol. 40, no. 7, pp. 1557–1565, Jul. 2005.

²Owing to non-idealities in device physical properties, in reality, the entropy contained in a cell is not maximized. In addition, β_s is normally higher than β_d because it is completely originated from the process variations of PCM cells, whereas β_d results from the collective effect of both device process variations and the characteristics of $Res_{target}^{\Gamma(j)}$, $V_{p,(1)}^{\Gamma(j)}$, and $t_{p,(1)}^{\Gamma(j)}$.