Effect of Aging on PUF Modeling Attacks based on Power Side-Channel Observations

Trevor Kroeger*, Wei Cheng†, Sylvain Guilley‡†, Jean-Luc Danger†‡ and Naghmeh Karimi*

*CSEE Department
University of Maryland Baltimore County
Baltimore, MD 21250
firstname.lastname@umbc.edu

†LTCI, CNRS, Télécom ParisTech
Université Paris-Saclay
75 013 Paris, France
firstname.lastname@telecom-paristech.fr

‡Think Ahead Business Line
Secure-IC S.A.S.
75 015 Paris, France
firstname.lastname@secure-ic.com

***Abstract*—Thanks to the imperfections in manufacturing process, Physically Unclonable Functions (PUFs) produce their unique outputs for given input signals (challenges) fed to identical circuitry designs. PUFs are often used as hardware primitives to provide security, e.g., for key generation or authentication purposes. However, they can be vulnerable to modeling attacks that predict the output for an unknown challenge, based on a set of known challenge/response pairs (CRPs). In addition, an attacker may benefit from power side-channels to break a PUFs' security. Although such attacks have been extensively discussed in literature, the effect of device aging on the efficacy of these attacks is still an open question. Accordingly, in this paper, we focus on the impact of aging on Arbiter-PUFs and one of its modeling-resistant counterparts, the Voltage Transfer Characteristic (VTC) PUF. We present the results of our SPICE simulations used to perform modeling attack via Machine Learning (ML) schemes on the devices aged from 0 to 20 weeks. We show that aging has a significant impact on modeling attacks. Indeed, when the training dataset for ML attack is extracted at a different age than the evaluation dataset, the attack is greatly hindered despite being performed on the same device. We show that the ML attack via power traces is particularly efficient to recover the responses of the anti-modeling VTC PUF, yet aging still contributes to enhance its security.**

***Index Terms*—Physically Unclonable Function; ISO/IEC 20897; Modeling Attacks; Side-Channel Attacks; Device Aging.**

I. Introduction

Physically Unclonable Functions (PUFs) are broadly deployed to provide a unique signature for integrated circuits and in turn enhance their security level. PUFs mainly represent one-way functions and accordingly they can be used for the authentication of Integrated Circuits (ICs), by using Challenge/Response Pair protocols or generation of a cryptographic key specific to the device. Using PUFs prevents storing secret information (e.g., cryptographic keys) in digital memory, thereby enhancing the security of the systems in which they are deployed [1]. Considering their small size, PUFs are broadly used in low-cost devices such as smart cards.

In practice, the unique behavior of each PUF instance (with similar gate-level implementation) is owed to the static randomness, related to unintended technological dispersions, that follows a normal distribution [2]. Due to such manufacturing imperfections, the PUF responses to input challenges are unique for each fabricated instance.

As PUFs play an important role to enhance the security of integrated circuits, they should be designed such that they cannot be cloned and their outputs cannot be predicted easily. One classification of PUFs relies on families: strong PUFs and weak PUFs. A weak PUF utilizes a limited number of Challenge/Response Pairs (CRPs), e.g. SRAM PUF [3], and is more suitable for key generation, while a strong PUF exhibits a large number of CRPs (e.g., arbiter-PUF [4]), and thereby is suitable for authentication [2]. Weak PUFs can be exposed to invasive attacks in which the internal structure of the PUF is manipulated or monitored [5], or to cloning attacks where the circuitry of the PUF is accurately replicated for similar responses [6]. Strong PUFs can be compromised with modeling attacks [7], [8], side-channel attacks [9], or a combination of the two [10], [11]. In the modeling attacks an adversary collects extensive number of CRPs and uses them to predict the PUF response for other challenges based on numeric methods including Machine Learning (ML) techniques.

In this paper, we consider the situation where the PUF can be profiled arbitrarily by a malevolent attacker during the enrollment phase just after tape-out. The profiling can be done by either using the CRPs or by using a "side-channel" like the power consumption traces of the PUF. These traces are of great interest for a ML attack as once the chip has been enrolled and the response channel is not accessible anymore (generally this channel is cut by an antifuse), the only way to observe the response is by side-channel. Therefore, one can imagine the ML attack scenario where the attacker registers a training dataset during enrollment and perpetrate the attack when the PUF is in use with unknown challenges. This attack, performed only by using power traces, can be performed for any application, when the PUF is used for CRP authentication protocol or to generate a cryptographic key.

Although the effect of combined modeling and side-channel attacks on PUF has been largely investigated in literature [7]–[11], the success of such attacks has only been studied on new (unaged) PUFs. However, the aging-induced changes in the transistors parameters over time can alter the accuracy of the attacks that model the PUFs' behaviors via their power traces. Accordingly, in this paper, we will focus on the impact of device aging (mainly Bias Temperature-Instability (BTI) and Hot-Carrier Injection (HCI) aging mechanisms) on such attacks and show that aging can hinder such attacks if the data used during the training phase of PUF modeling is gathered at a different time than the actual time the PUF is attacked. It is noteworthy to mention that normally a PUF is not solicited very often (e.g., solicitation occurs at each boot,

978-3-9819263-4-7/DATE20/©2020 EDAA

or upon each authentication), PUF aging may be considered a secondary issue. However, for the following reasons, in industrial products, a PUF is used more frequently [1]:

- There is usually a BIST (Built-In Self-Test) at each system power-up;
- In a view to improve its reliability, a PUF is typically called several times to vote which bit is the most stable; hence the most likely (despite the noise).

Besides, a malicious user may compromise a PUF by requesting continuous authentications.

In this paper, we target not only the arbiter-PUF which has been already shown to be vulnerable to modeling attacks [8], but also the Voltage Transfer Characteristic (VTC) PUF which is designed as an arbiter-PUF counterpart to resolve the modeling effect of arbiter-PUF [12]. We investigate the effect of aging on the resiliency of these two PUFs against modeling attacks that use the power side-channel to model them. In addition, we show that the VTC PUF which is supposed to be resilient against modeling attacks, can be broken via ML attacks using power side-channel. The contributions of this paper are as follows:

- Success rate results of ML attacks against arbiter and VTC PUFs according to: ML algorithm, ML input (CRPs, power traces), PUF size (16,64), and training dataset size;
- Successful attack of anti-modeling VTC PUFs via ML algorithms using power traces;
- HSpice MOSRA simulations to evaluate the effect of NBTI and HCI aging mechanisms on the success of the ML attacks on PUF;
- Analysis showing that the security against ML attacks can profit from aging;

The rest of this paper is organized as follows. Section II discusses the threat model considered in this paper. Section III presents the preliminary background on aging mechanisms and describes the PUFs we target in this study. Modeling attacks using ML algorithms are also briefly discussed in this section. Section IV presents the simulation results and discussions. Finally, conclusions and future extensions of this research are drawn in Section V.

II. Attacking PUFs

A. Threat Model

In this paper, we target the arbiter-PUF and its modeling resistant counterpart the so-called VTC PUF (proposed in [12]) and launch two modeling attacks on them. The first attack uses a set of known CRPs from each target PUF to model its behavior, while the second attack deploys power side-channel to break the security of the targeted PUFs. We investigate the success of the second attack when the training and target data sets are not aligned in terms of aging, i.e. the PUFs are attacked at time t using the data gathered at time where $t_0 \neq t$. However, *we don't consider the effect of aging on the first attack (i.e., modeling based on CRPs)*, since aging doesn't have impact on challenges (namely, aging will not change the bit-string into a PUF), but affects physical properties like power consumption, Electro-Migration radiation, etc. We can perform ML for unknown CRPs, but their inference after learning is not affected by aging (unless reliability drops, which is beyond the scope of this paper; for those cases a number of techniques are taken, e.g. pruning of challenges, to enhance reliability and withstand aging).

Typically, the chip is modeled when it is still "open" (enrollment phase) to play any challenges. The attack occurs subsequently, to retrieve valuable secrets (in post-customization phase).

B. Modeling Attacks based on Challenge and Response Pairs

Typically modeling attacks are based on challenge and responses of a PUF. The goal of this attack is to glean the response from the challenge. In practice, the modeling attack attempts to characterize F in the equation: $r = F(c)$ where r is the response set and c is the challenge set. A large number of challenges and their corresponding responses are collected and used to train a model that mimics the behavior of the PUF. The attacker can then predict the response of the PUF to any challenge given to this device as if it is the original device. The modeling is performed by taking advantage of ML algorithms [7], [8].

C. Modeling Attacks based on Power Traces

By monitoring the current drawn by the PUF, the underlining characteristics of the PUF's circuitry can be discerned. To perform this attack, in practice, the power traces of a PUF are monitored to model the PUF [10], [13], [14]. This is done by sampling the current of the PUF when it is queried with a challenge. As the input edge propagates through the PUF (e.g., in the arbiter-PUF) the traces are sampled at consistent intervals. These traces are correlated to the physical specification of the targeted PUF and can be used (instead of challenges) to train a ML model to mimic the PUF behavior. This model can then used to infer the PUF responses.

III. Preliminary Background

A. Background on Aging

Aging mechanisms result in performance degradation and eventual failure of digital circuits over time. The two leading factors in CMOS technology are Negative Bias Temperature-Instability (NBTI) and Hot-Carrier Injection (HCI) [15]. Both aging sources result in increasing switching and path delays.

NBTI Aging: NBTI affects a PMOS transistor when a negative voltage is applied to its gate. A PMOS transistor experiences two phases of NBTI depending on its operating condition. The first phase, the so-called stress phase, occurs when the transistor is on ($V_{gs} < V_t$). Here, positive interface traps are generated at the Si-SiO$_2$ interface which lead to an increase of the threshold voltage of the transistor. The second phase, the so-called recovery phase, occurs when the transistor is off ($V_{gs} > V_t$). The threshold voltage drift that occurred during the stress phase will partially recover in the recovery phase. Threshold voltage drifts of a PMOS transistor under stress depend on the physical parameters of the transistor, supply voltage, temperature, and stress time [16]. Figure 1 shows the threshold voltage drift of a PMOS transistor that is continuously under stress for 6 months and a transistor that alternates stress/recovery phases every other month. As shown, the NBTI effect is high in the first couple of months but the threshold voltage tends to saturate for long stress times.

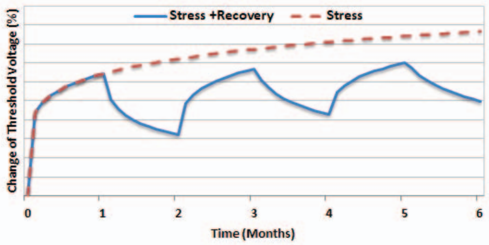


Fig. 1. Threshold-voltage shift of a PMOS transistor under NBTI effect.[1]

HCI Aging: HCI occurs when hot carriers are injected into the gate dielectric during transistor switching and remain there. HCI is a function of switching activity and degrades the circuit by shifting the threshold voltage and the drain current of transistors under stress. HCI mainly affects NMOS transistors. HCI-induced threshold voltage drift is highly sensitive to the number of transitions occurring in the gate input of the transistor under stress. In practice, HCI has a sublinear dependency on the clock frequency, usage time, and activity factor of the transistor under stress, where activity factor represents the ratio of the cycles the transistor is switching and the total number of cycles the device is utilized. HCI effects depend on the operating temperature [15].

B. Background on Arbiter-PUF

An arbiter-PUF is composed of a pair of delay chains and generates one response bit per challenge, in one single query [17]. In practice, this PUF operates based on the process-variation induced race between two identical paths (top and bottom paths shown in Fig. 2). The race corresponds to the difference of the delay of these two paths, and is grabbed by the arbiter [4]. In fact, only the sign of this difference is important (not the exact amount). The sign, which is extracted by the arbiter, presents the PUF identifier (response). The arbiter can be realized as simple S-R latch implemented by two NAND gates.

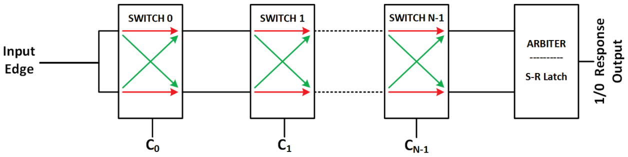


Fig. 2. Structure of an Arbiter-PUF [4].

C. Voltage Transfer Characteristic (VTC) PUF

The VTC PUF was initially proposed in [12] to address the weakness of some strong PUFs (mainly the arbiter-PUF) against modeling attacks. As shown in Fig. 3, a VTC PUF includes non-linear blocks between the switches to increase the resistance of the underlying PUF against modeling attacks.

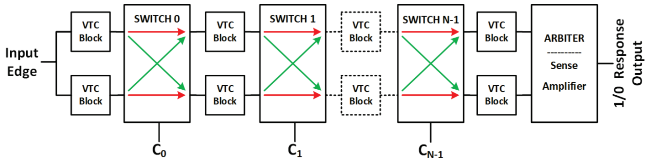


Fig. 3. Structure of a VTC PUF [12].

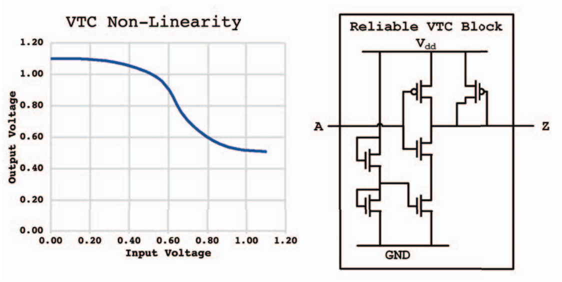


Fig. 4. Phenomenon exploited by the VTC PUF (left) that is produced by the VTC Circuit (right) [12].

The VTC PUF operates with analog voltage levels. The challenge bits are provided to the switches as in the arbiter-PUF, however this PUF is fed with a transition with the maximum level of $Vdd/2$ (to induce non-linearity) that propagates through the VTC Blocks (Fig. 4), as well as transmission-gate based switches (Fig. 5). Then the output voltages of the upper and lower last stage VTC blocks are compared to determine the PUF response [12].

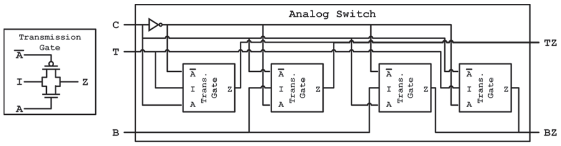


Fig. 5. Circuit diagram for the analog switch used in the VTC PUF.

D. Machine Learning/Modeling Attack

The modeling, performed with ML algorithms, consists of two phases: training and evaluation (or inference). In the training phase the model is constructed utilizing a set of inputs, in this paper CRPs or power traces of the target PUFs, and their corresponding responses. The model is adjusted based on whether it classifies the input to the correct response or not. In the evaluation phase, unseen inputs (again these are the challenges or the power traces) are tested to see if the model correctly classifies the output. In this paper we use the Support Vector Machine (SVM) [18], Decision Tree [19], and Random Forest [20] algorithms to launch the modeling attacks.

IV. Experimental Results and Discussion

A. Experimental Setup

In this research, we implemented the targeted arbiter-PUF and VTC PUF at the transistor level using a 45 nm technology extracted from the open-source NANGATE library [21]. We then used Synopsys HSpice for the transistor-level simulations

[1] Values on Y axis are not shown to make the graph generic for different technologies.

and deployed the HSpice built-in MOSRA Level 3 model [22] to capture aging effects. We evaluated the effect of both BTI and HCI effects for 20 weeks of PUF operation in time steps of one week, with a temperature of 80°C. We mainly performed our analysis on 16-stage PUFs. However, for a fair comparison with [12], we also implemented a 64-bit VTC PUF and investigated attacking that PUF using power traces. We deployed three different ML algorithms to model each PUF, mainly SVM, Decision Tree and Random Forest algorithms.

All modeling experiments were performed on a PC on a quad-core processor (Intel Core i5-7200U) running at 2.50 GHz with 16 GB of memory. The training time of the SVM algorithm is shown in Table I for the three targeted PUFs (by using CRPs and power traces).

Data Extraction: Figure 6 shows the timing considered for the data extraction from the targeted PUFs. The PUF is fed with a rise transition 2.5ns after applying each challenge. The PUF response is extracted after it becomes stable. To extract power traces for the attack, the circuit current is sampled 1000 times between the time that the PUF is fed with the rise transition and the time that the response becomes stable. Figure 7 shows a set of collected traces, sampled within the aforementioned window, for both PUFs investigated.

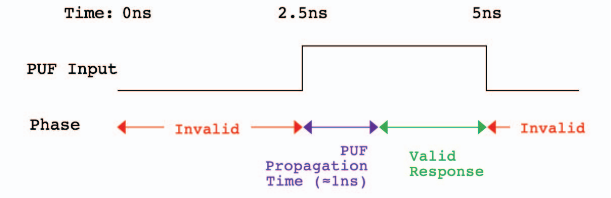


Fig. 6. Power trace and response sampling window.

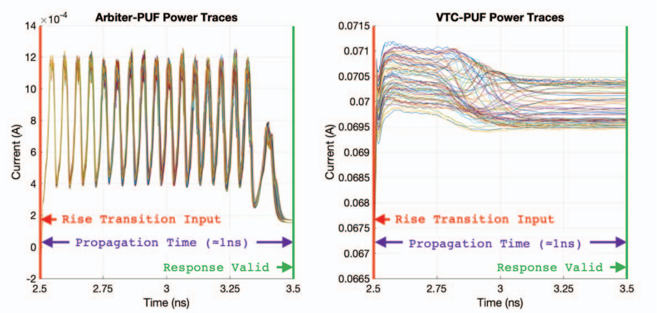


Fig. 7. Power traces collected for the PUFs in the propagation time window.

B. Experimental Results

In this section, we present our results and discuss our observations.

1) Modeling attacks on Fresh PUFs: The first set of results investigates the success of the modeling attacks on both targeted PUFs when the PUFs are fresh (unaged). The results are shown in Table I. In these experiments, 2000-8000 random CRPs/power traces have been used to train the models (based on the SVM algorithm) for each PUF and the models were tested against 6000 random CRPs/power traces.

Our results for attacking the 16-bit arbiter-PUF when CRPs are used for modeling follow those in literature [8], [23], i.e. arbiter-PUFs can be accurately modeled via a subset of their CRPs. However, the results for modeling the 16-bit VTC PUF using CRPs are different from literature [12] since the results in Table I show that the 16-bit VTC PUF can be modeled via CRPs. However, it is noteworthy to mention that the results reported here are for a 16-bit PUF using 4000 CRPs, i.e. 6.1% of the total number of possible challenges were used in this experiment which is more than a realistic case. Thereby, to have a fair comparison with [12], we also implemented a 64-bit version. The results of modeling that PUF via the SVM algorithm are also shown in Table I. As shown, the accuracy in predicting the PUF response is 53.23% for 4000 CRPs, i.e. the VTC PUF cannot be easily modeled using its CRPs.

Although the attacker may not benefit from using CRPs in modeling VTC PUFs, we show that these PUFs are highly vulnerable to the attacks using the power side-channel. The results in Table I for both 16-bit and 64-bit VTC PUFs confirm our claim that the VTC PUF can be modeled utilizing power traces even though this PUF is supposed to be resistant against modeling attacks based on CRPs [12]. On the other hand, Table I also shows that the arbiter-PUF can be easily modeled via its power traces, as expected. Please note that as the arbiter-PUF is known to be vulnerable to modeling attacks [8], [23], we do not present the results for the 64-bit variant of this PUF due to the limited space.

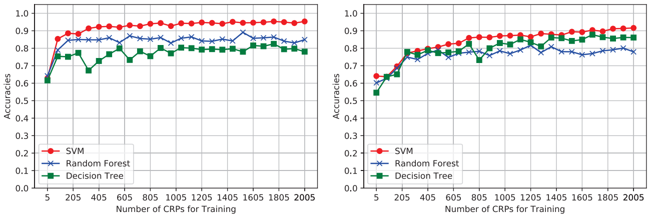
The takeaway point from these observations is that VTC PUF can be modeled via its power traces. *To the best of our knowledge, this is the first successful modeling of this PUF*. Also as shown and expected based on literature, the arbiter-PUF can be modeled easily via its CRPs or its power traces.

TABLE I
MODELING ATTACKS ON 16-BIT VTC PUF AND ARBITER-PUF, AND 64-BIT VTC PUF BY USING CRPS AND POWER TRACES.

PUFs	Data	Number of Training Data	Accuracies	Training Time (s)
16-bit VTC PUF	CRPs	2000	0.9493	0.0767
		4000	0.9603	0.1718
		8000	0.9657	0.5241
	Power Traces	2000	0.9510	0.5933
		4000	0.9612	1.8322
		8000	0.9677	5.5863
16-bit arbiter-PUF	CRPs	2000	0.9117	0.166
		4000	0.9485	0.4813
		8000	0.9547	1.5797
	Power Traces	2000	0.9460	0.9655
		4000	0.9545	2.9757
		8000	0.9665	8.8265
64-bit VTC PUF	CRPs	2000	0.5173	0.5280
		4000	0.5323	2.1631
		8000	0.5223	17.323
	Power Traces	2000	0.8303	2.1868
		4000	0.8428	7.9662
		8000	0.8535	31.5004

2) PUF Model Training and Accuracy: The next set of results investigates the success of the modeling attacks based on the number of CRPs used to train the model. In both experiments shown in Fig. 8, between 5 to 2000 CRPs were used to generate the model in each attack. As depicted, for both PUFs, the SVM algorithm performs better at modeling the 16-bit targeted PUFs than the Random Forest and Decision Tree algorithms. The results show that the targeted arbiter-PUF is modeled with the accuracy of 90% with ≈300 traces, while the VTC PUF needed ≈1600 traces for the same accuracy level. This is due to the non-linearity of the logic blocks in the VTC PUF. As mentioned in Section IV-B1, modeling of VTC PUF using CRPs is successful due to the small size of PUF (16 stages), while arbiter-PUF can be easily modeled even with greater number of stages.

Although the accuracy of the utilized models tend to increase with the number of CRPs used for training, after applying ≈400 CRPs, there is no significant increase in the accuracy achieved for modeling the arbiter-PUF. Similar observation can be made for the VTC PUF (with ≈1800 traces).

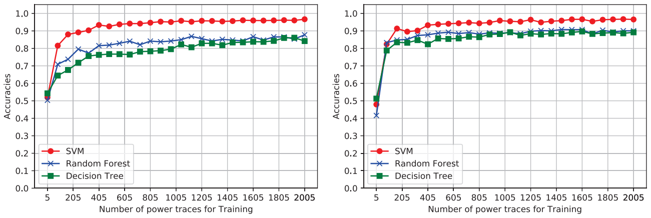


(a) 16-Bit arbiter-PUF.

(b) 16-Bit VTC PUF.

Fig. 8. Accuracies achieved for different number of CRPs used in Training.

Figure 9 presents the number of power traces required to successfully model each PUF. For these experiments, the model was generated with 5 to 2000 power traces. As depicted, for the arbiter-PUF the accuracies are quite similar to those for CRPs (Fig 8a). However, SVM outperforms Decision Trees and Random Forest by at least 5% accuracy. For the VTC PUF, SVM still outperforms, however its results are much closer to the other two ML algorithms. For both PUFs a high level of accuracy, over 90%, is achieved at 320 traces for the arbiter-PUF and even less for the VTC PUF with ≈160 traces. Stability is achieved for both PUFs at ≈600 traces with a ≈95% accuracy.

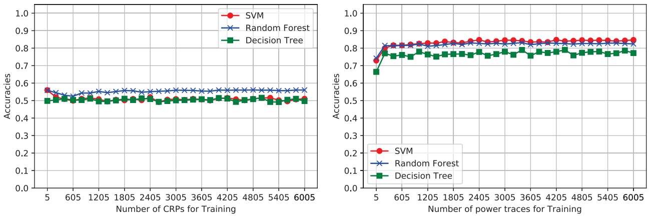


(a) 16-Bit arbiter-PUF.

(b) 16-Bit VTC PUF.

Fig. 9. Accuracies achieved for different number of power traces used in Training.

Figure 10 depicts the accuracy versus the number of training traces for the 64-bit VTC PUF. The accuracy is reported based on a test set of 6000 CRPs (Fig. 10a) and 6000 power traces (Fig. 10b). These results confirm our assumption that modeling large size VTC PUFs is difficult using CRPs. As shown in Fig. 10a, the accuracy stays constant at ≈50%-55% regardless of the number of traces (even with 60,000 CRPs for training). However, the VTC PUF can be modeled with an accuracy of ≈85% with only 600 traces (out of 2^{64} possible traces). The takeaway point from these observations is that VTC PUF is not resilient against modeling attacks based on power traces.



(a) Using CRPs.

(b) Using power traces.

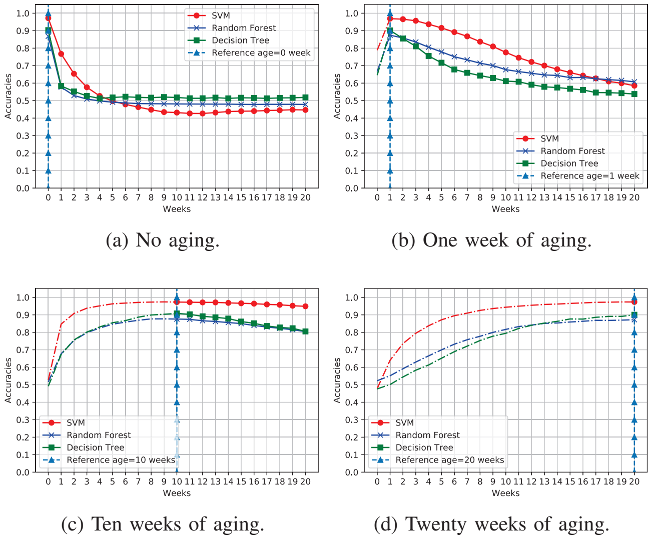
Fig. 10. Modeling accuracies for the 64-Bit VTC PUF based on the number of CRPs and power traces used to train the model.

3) Attacking Aged Arbiter-PUFs: This set of results focuses on the aging of the arbiter-PUF and how it affects our ability to model it. Figure 11 shows the accuracy of the attack when the targeted device has been used between 1 and 20 weeks, while the training data was gathered from a new, 1-week, 10-week, or 20-week old device. The training and evaluation sets included 4000 and 6000 power traces, respectively.

Figure 11a depicts the case in which a new arbiter-PUF was used for training. As shown, depending on the deployed ML algorithm, with approximately 85%-95% accuracy, a new device (unaged, 0-week) can be attacked using the power traces gathered from the device at that time. However, when the device has been used for some time, it can not be easily attacked using the power traces gathered at another time. For example, as shown in Fig. 11a, we cannot attack an arbiter-PUF after 20 weeks of aging (accuracy≈50%). In practice, these results show that to improve the accuracy of the attack, the target device should be modeled via traces that have been gathered at the same or near age. All graphs in Fig. 11 follow the same trend. For example, by using the SVM algorithm, a model for a 1-week old arbiter-PUF will have an accuracy of ≈60% if deployed on a device that has been aged to 20 weeks, while, the attack accuracy increases to 95% if the training data has been extracted at same age. i.e. 1-week.

The first takeaway point from this result is that, as aging is always more dominant in the first weeks of device operation, the effect of misalignment between the age of the training and modeling traces results in higher attack difficulty in the first weeks of device operation, i.e. the modeling accuracy has a faster drop due to the misalignment of aging duration when the training data is extracted from a new (Fig. 11a) or 1-week old device (Fig. 11b) as compared to the other ages shown in Fig. 11c and Fig. 11d. The second takeaway point is that aging hinders the modeling attacks on arbiter-PUFs if the training data has been gathered previously, for example when the device was new (or at time t_0), and there is no physical access to the device to get the power traces at time $t > t_0$.

4) Modeling of Aged VTC PUFs: The fourth set of results presents the effect of aging in the modeling of VTC PUFs using power side-channel. The results are shown in Fig. 12. The observations are very similar to those of the arbiter-PUF. In practice, the misalignment in age of the targeted PUF when



(a) No aging.

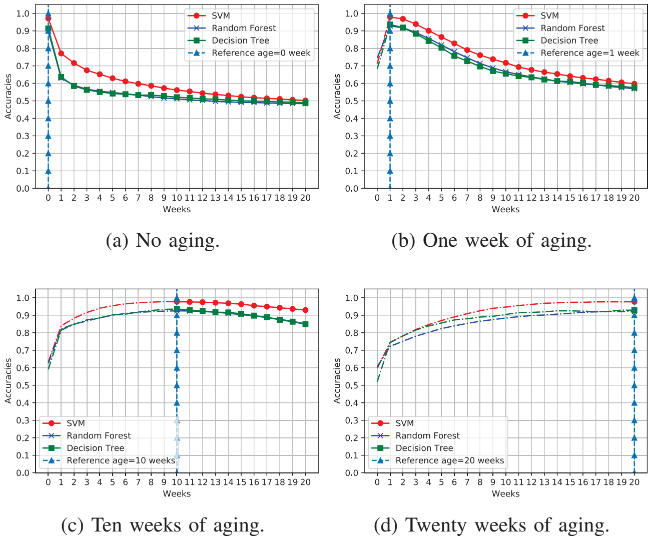
(b) One week of aging.

(c) Ten weeks of aging.

(d) Twenty weeks of aging.

Fig. 11. Accuracy for modeling attacks on 16-bit arbiter-PUFs (aged between 0 and 20 weeks) when the training model was extracted from an X-week old device, where X$\in \{0, 1, 10, 20\}$. The portions shown with dashed lines represent the cases in which the attacker uses the current traces of the PUF to extract the previous PUF responses (e.g., extracting previous keys in key-management schemes to analyze previously stored encrypted data.)

it is attacked versus when its power traces were extracted hinders the modeling attack. In particular, as Fig. 12a shows, the model trained with the power traces extracted from a new device, cannot predict the output correctly if the device has been used for a few weeks (accuracy is less than 60% after 7 weeks). Similar observations can be made for the cases in which the training data has been extracted for a 1-week, 10-week, or 20-week old device (Fig. 12b-12d).



(a) No aging.

(b) One week of aging.

(c) Ten weeks of aging.

(d) Twenty weeks of aging.

Fig. 12. Accuracy for modeling attacks on 16-bit VTC PUFs (aged between 0 and 20 weeks) when the training model was extracted from an X-week old device, where X$\in \{0, 1, 10, 20\}$. The portions shown with dashed lines represent the cases in which the attacker uses the current traces of the PUF to extract the previous PUF responses (e.g., extracting previous keys in key-management schemes to analyze previously stored encrypted data.)

V. Conclusion and Future Directions

In this paper, we studied the modeling attacks on strong PUFs by using power traces, in order to analyze the impact of aging on modeling the arbiter-PUF and its anti-modeling counterpart, the VTC PUF. We showed that although the VTC PUF cannot be modeled via CRPs, its functionality can be revealed through its power side-channel. In addition, we showed that aging misalignment between the training and attacking phases will hinder the effectiveness of the attack. In the continuation of this work, we will investigate our findings in PUFs realized in real silicon.

Acknowledgement

The authors would like to thank Professor Sandip Kundu and Vinay Patil from the University of Massachusetts Amherst for providing technical assistance about the VTC PUF.

References

[1] N. Karimi, J.-L. Danger, and S. Guilley, “Impact of aging on the reliability of delay PUFs,” *JETTA*, vol. 34, no. 5, pp. 571–586, 2018.

[2] C. Herder, M. Yu, F. Koushanfar, and S. Devadas, “Physical unclonable functions and applications: A tutorial,” *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1126–1141, 2014.

[3] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, “FPGA intrinsic PUFs and their use for IP protection,” in *CHES*, 2007, pp. 63–80.

[4] B. Gassend, D. E. Clarke, M. van Dijk, and S. Devadas, “Silicon physical random functions,” in *CCS*, 2002, pp. 148–160.

[5] D. Nedospasov, J. Seifert, C. Helfmeier, and C. Boit, “Invasive PUF analysis,” in *FDTC*, 2013, pp. 30–38.

[6] C. Helfmeier and C. Boit, “Cloning physically unclonable functions,” in *HOST*, 2013, pp. 1–6.

[7] U. Rührmair and J. Sölter, “PUF modeling attacks: An introduction and overview,” in *DATE*, 2014, pp. 1–6.

[8] U. Rührmair et al., “Modeling attacks on physical unclonable functions,” in *CCS*, 2010, pp. 237–249.

[9] D. Merli et al., “Side-channel analysis of PUFs and fuzzy extractors,” in *Trust and Trustworthy Computing*, 2011, pp. 33–47.

[10] A. Mahmoud et al., “Combined modeling and side channel attacks on strong PUFs,” *Cryptology ePrint Archive*, 2013. [Online]. Available: http://eprint.iacr.org/2013/632

[11] U. Rührmair et al., “Efficient Power and Timing Side Channels for Physical Unclonable Functions,” in *CHES*, 2014, pp. 476–492.

[12] A. Vijayakumar and S. Kundu, “A novel modeling attack resistant PUF design based on non-linear Voltage Transfer Characteristics,” in *DATE*, 2015, pp. 653–658.

[13] G. T. Becker and R. Kumar, “Active and passive side-channel attacks on delay based PUF designs,” *IACR Cryptology Archive*, vol. 2014, p. 287, 2014. [Online]. Available: https://eprint.iacr.org/2014/287

[14] K. Fukushima et al., “Delay PUF assessment method based on side-channel and modeling analyzes: The final piece of all-in-one assessment methodology,” in *IEEE Trustcom/BigDataSE/ISPA*, 2016, pp. 201–207.

[15] F. Oboril and M. B. Tahoori, “Extratime: Modeling and analysis of wearout due to transistor aging at microarchitecture-level,” in *DSN*, 2012, pp. 1–12.

[16] S. Khan et al., “NBTI monitoring and design for reliability in nanoscale circuits,” in *DFTS*, 2011, pp. 68–76.

[17] G. E. Suh and S. Devadas, “Physical unclonable functions for device authentication and secret key generation,” in *DAC*, 2007, pp. 9–14.

[18] S. Yue, P. Li, and P. Hao, “SVM classification: Its contents and challenges,” *Applied Mathematics-A Journal of Chinese Universities*, vol. 18, no. 3, pp. 332–342, Sep 2003.

[19] S. B. Kotsiantis, “Decision trees: a recent overview,” *Artificial Intelligence Review*, vol. 39, no. 4, pp. 261–283, Apr 2013.

[20] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct 2001.

[21] “Nangate 45nm open cell library,” “http://www.nangate.com” (accessed May 2019).

[22] Synopsys, “HSPICE User Guide: Basic Simulation and Analysis,” 2016.

[23] U. Rührmair et al., “PUF modeling attacks on simulated and silicon data,” *IFS*, vol. 8, no. 11, pp. 1876–1891, 2013.