Coin Flipping PUF: A Novel PUF with Improved Resistance against Machine Learning Attacks

Yuki Tanaka, *Student member, IEEE,* Song Bian, *Student member, IEEE,*
Masayuki Hiromoto, *Member, IEEE,* and Takashi Sato, *Member, IEEE*

***Abstract*—We propose a novel coin-flipping physically unclonable function (CF-PUF) that significantly improves the resistance against machine-learning attacks. The proposed PUF utilizes the strong nonlinearity of the convergence time of bistable rings (BRs) with respect to variations in the threshold voltage. The response is generated based on the instantaneous value of a ring oscillator at the convergence time of the corresponding BR, which is running in parallel. SPICE simulations show that the prediction accuracy of support-vector machine (SVM) on the responses of CF-PUF is around 50 percent, which means that SVM cannot predict better than random guesses.**

***Index Terms*—PUF, Hardware Security, Machine Learning, Ring Oscillator, Bistable Ring.**

I. Introduction

Physically unclonable functions (PUFs) [1], [2] are attracting increasing attention in the field of hardware security. PUFs utilize the inherent physical variations of hardware components to generate chip-specific secret keys. The PUF circuit serves as the function, $r = f_\alpha(\boldsymbol{c})$, that returns a response, r, upon receiving a challenge input $\boldsymbol{c}$. Depending on the physical variations in hardware, which are represented by α, a unique and unclonable set of challenge-response pairs (CRPs), $(\boldsymbol{c}, r)$, is determined for each PUF chip. A variety of PUFs have been proposed to ensure security, including the SRAM PUF [3], arbiter PUF [1], ring-oscillator PUF (RO-PUF) [2], and bistable-ring PUF (BR-PUF) [4].

The performance of the PUFs has traditionally been evaluated using the metrics such as uniqueness and reliability [5]. However, with the recent advances in machine learning (ML), the ability of PUFs to resist ML attacks has been scrutinized [6]–[8]. Generally, in an ML attack, the attacker tries to characterize the entire CRP space by observing the ML-based inference and conducting learning on a small set of CRPs. A secure PUF needs to be immune to such ML attacks.

It is known that some of the existing PUFs, such as arbiter PUFs and BR-PUFs, can be modeled by simple linear functions [6]–[8], which makes it trivial to predict the PUF response by using ML algorithms. In a previous study [8], support-vector machine (SVM) attacks were used to predict the responses of BR-PUF and variants of BR-PUF, with a success rate higher than 95%.

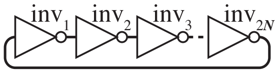


Fig. 1. Circuit structure of a BR.

Here, our objective is to develop a novel PUF architecture that is resilient against ML attacks. We start by analyzing the convergence time of BR circuits. Through analytical formulations, the convergence time of a BR is shown to be nonlinearly dependent on the variable source, which is associated with the variation in the threshold voltage of the transistors. Leveraging this nonlinearity, the *coin-flipping PUF* (CF-PUF) is proposed, which includes an RO and a BR, and its response is determined by the instantaneous value of the RO at the exact time when the BR paired to it converges. Extensive SPICE simulations have shown that the proposed CF-PUF maintains an ideal 50% prediction accuracy against SVM attacks.

II. Preliminaries

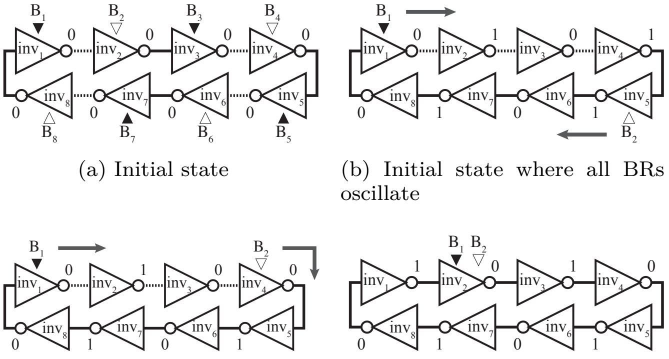
A. Bistable Ring

A BR is a ring formed by an even number of inverters with a circuit structure as shown in Fig. 1. It has two stable states that are the clockwise enumerations of the inverter outputs starting from inv$_1$: "$0101 \cdots 01$" (for which the first four digits in hexadecimal notation correspond to "5") or "$1010 \cdots 10$" (corresponding "A"). If an initial value other than these two stable states is given, the BR starts to oscillate until it reaches either of the two steady states. Because the stable states are determined by the physical variations of the transistors that compose the inverters, BRs can be used to generate a chip-specific response.

Based on Fig. 2, we consider the convergence of the BR in detail. When "0" is input to all inverters, as shown in Fig. 2(a), all the inverters lie at the boundaries between "5" and "A," which are indicated by B$_i$ in the figure. As a signal propagates through the inverters, these boundaries also propagate. During this period, the output of a specific inverter becomes "0" and "1" repeatedly, meaning that the BR is oscillating. Over time, the distance between

Y. Tanaka, S. Bian, M. Hiromoto, and T. Sato are with Department of Communications and Computer Engineering, School of Informatics, Kyoto University, Japan (correspondence e-mail: paper@easter.kuee.kyoto-u.ac.jp).

This work was supported in part by Japan Society for the Promotion of Science (JSPS) KAKENHI Grant Number 17H01713. This work was also supported by VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Synopsys, Inc.



(a) Initial state

(b) Initial state where all BRs oscillate

(c) Oscillation state

(d) Convergence state

Fig. 2. Oscillation and convergence states of a BR.

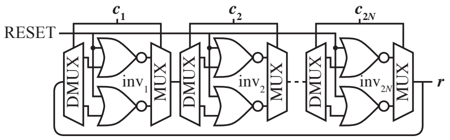


Fig. 3. Circuit structure of a BR-PUF.

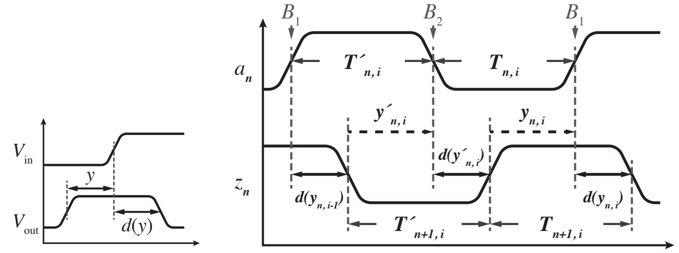
the boundaries diminishes because of the difference in the propagation speed of each boundary caused by variations in the inverter delays. As shown in Fig. 2(c), when one boundary catches up to another, the number of boundaries in the BR decreases. Eventually, the system reaches a convergence state when none of the boundaries remain, as shown in Fig. 2(b).

This observation indicates that the initial value given to the BR is important to control the number of boundaries and their positions. Different initial conditions result in different oscillation times. If the initial value is as shown in Fig. 2(b), the distance between the boundaries is maximized, and the oscillation time will be longer. In the proposed circuit, this property is utilized to lengthen the convergence time of the BR.

B. Bistable Ring PUF

In practical BR-PUFs, all the inverters are realized by two dual-input NOR gates and a multiplexer (MUX) / demultiplexer (DMUX) pair that selects one of the NOR gates. The select signal is used as a challenge, and the output of one of the MUXs is used as the response. Since the delay of the NOR gate varies, different BRs return different responses to the challenges. The other input pin of each of the NOR gates is used for a reset signal and is referred to as RESET. If RESET=1, the outputs of all the NOR gates become 0 regardless of the input values. On the other hand, if RESET=0, the BR-PUF operates as a BR so the initial value for all the inverters is 0.

A response to a challenge is obtained by first setting RESET=1 for initialization and then setting RESET=0. The output after the BR becomes stable is used as the response.



(a) Propagation in one inverter.

(b) Signal propagation in the n-th inverter.

Fig. 4. Delay propagation of inverters in BR.

III. Convergence Property of a BR

A. Formulation of the Oscillation Behavior of a BR

Based on the drafting effect [9], [10], the propagation delay d of the inverter in Fig. 4(a) can be approximated as a function of the time y elapsed after the last commutation of its output:

$$d(y) = D - \alpha e^{-\frac{y}{\tau}}, \tag{1}$$

where D is a linear function of the threshold voltage when y is sufficiently large, and α and τ are constants. Let D_{p} and D_{n} denote the functions D in Eq. (1) when the output of the inverter is rising and falling, respectively. The changes in the delay of the inverter can be approximated by a linear function of the variations in the threshold voltage [11]. Therefore, D_{p} and D_{n} can also be expressed as linear functions of the threshold voltages of the pMOS and nMOS transistors, respectively.

Consider the case in which an initial value is given to the $2N$-stage BR as in Fig. 2(b). The input and output waveforms of the n-th inverter, a_n and z_n, respectively, are shown in Fig. 4(b). In this figure, $T_{n,i}$ (or $T'_{n,i}$) represents the time from boundaries B_1 to B_2 (or from B_2 to B_1), after the edge of B_1 passes the n-th inverter i times. From (1), $T_{n,i}$ can be expressed based on the propagation delays, $d(y_{n,i})$ and $d(y'_{n,i})$, as follows:

$$\begin{aligned} T_{n+1,i} &= T_{n,i} + \{d_n(y_{n,i}) - d_n(y'_{n,i})\} \\ &= T_{n,i} + \{D_{\mathrm{n},n} - D_{\mathrm{p},n}\} - \{\alpha e^{-\frac{y_{n,i}}{\tau}} - \alpha e^{-\frac{y'_{n,i}}{\tau}}\}, \end{aligned} \tag{2}$$

where $D_{\mathrm{p},n}$ and $D_{\mathrm{n},n}$ denote D_{p} and D_{n} for the n-th inverter, respectively. $T'_{n,i}$ can be modeled using an equation of the same form.

Thus, considering a_1 as the response, R_{BR}, the convergence time can be calculated as $T_{\mathrm{BR}} = \sum_{i=1}^{\infty}(T_{1,i} + T'_{1,i})$. Since $T_{1,i}$ represents the time during which a_1 remains 0, R_{BR} becomes 1 as $T_{1,i}$ gradually becomes smaller and finally attains a width of 0. Meanwhile, if $T'_{1,i}$ becomes smaller and converges to 0, then R_{BR} becomes 0.

B. Validation of the Approximation Equations

Here, the discussion above is validated through SPICE simulations using a commercial SPICE simulator [12] and a commercial 65 nm process library. Ten thousand instances of the 16-stage BR were generated while assuming a Gaussian distribution for the threshold voltages.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSII.2018.2821267, IEEE Transactions on Circuits and Systems II: Express Briefs

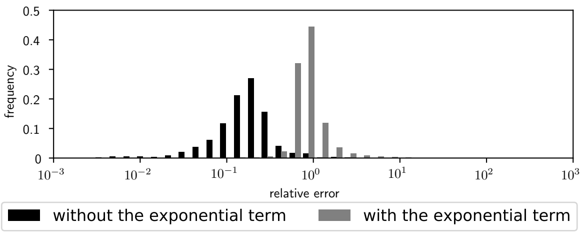


Fig. 5. Distribution of the relative errors in the convergence time with and without the approximation.

First, $D_{\mathrm{p},n}$ and $D_{\mathrm{n},n}$ were calculated using the given threshold voltage variations. Next, the relationship between D and the threshold voltage were obtained through SPICE simulations, given the parameters such as α and τ. Then, R_{BR} and T_{BR} were obtained using the equations in Sec. III-A. To evaluate the impact of the exponential term, R_{BR} and T_{BR} were also calculated using only the linear component in Eq. (1), omitting the exponential term, $-\alpha e^{-\frac{y}{\tau}}$. Finally, the calculated R_{BR} and T_{BR} were compared with the results of the full SPICE simulation for each instance.

The error rate for the estimation of the response, R_{BR}, was 1.66% and 1.65% when conducted with and without the exponential term, respectively, which are low. Since D is linear with respect to the threshold voltage variation, R_{BR} can be approximated by a linear equation.

The distribution of the relative errors of convergence time, T_{BR}, is shown in Fig. 5. The average and the maximum errors in the T_{BR} estimation were 22.9% and 1650%, respectively, when the exponential term was included; these values were 161% and 167900%, respectively, when the exponential term was disregarded, which are 7.01X and 102X larger than those when the exponential term was included. Thus, it is concluded that the exponential term in Eq. (1), $-\alpha e^{-\frac{y}{\tau}}$, has a strong influence on the estimation of the convergence time of the BR.

The above results indicate that the *convergence value* can be accurately predicted by a linear function of the threshold voltages; in contrast, the *convergence time* is nonlinear with respect to the threshold voltage variation and an exponential term must be used to calculate it. Hence, we hypothesize that although PUFs utilizing the convergence value are vulnerable to ML attacks, their resistance to ML attacks can be greatly enhanced if the convergence time is used for generating the response of the PUF. Therefore, we propose a PUF that provides a response based on the convergence time of a BR.

IV. Coin Flipping PUF

A. Circuit structure of CF-PUF

The proposed CF-PUF utilizes the oscillation time of a BR to be more resistant to ML attacks. Fig. 6 shows the structure of CF-PUF consisting of two ring circuits and one flip-flop. One ring is a $2N$-stage BR, and the other is a $(2N+1)$-stage RO. They run simultaneously, and the instantaneous response of the RO at the convergence time of the BR is used as the response of CF-PUF. To lengthen the convergence time of the BR, the two farthest inverters at the Nth and the $2N$th stages are replaced with NAND and NOR gates when N is an even number, or two NORs when N is an odd number. The remaining inputs of the NOR or NAND gate are used for the reset signal. The D flip-flop captures the instantaneous value of the RO at the last rising edge of the BR prior to convergence, which becomes the response of CF-PUF. The name of the proposed CF-PUF is derived from the way in which the value of the RO, which is oscillating rapidly between 0 and 1, is captured by the BR as in a coin toss.

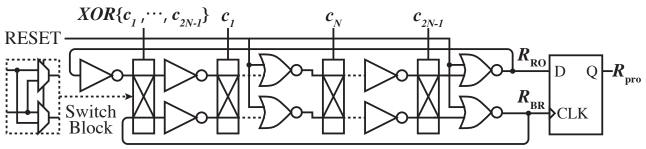


Fig. 6. Schematic of the proposed CF-PUF ($N \bmod 2 = 1$).

In CF-PUF, the BR and RO are interleaved with switch blocks that are in between the stages that comprise two MUXs to propagate the signal, which is either straight or crossed. Thus, the route selection of each switch block becomes a challenge input of CF-PUF. The total number of selection circuits is $2N$. However, if $\mathrm{XNOR}\{C_1, \cdots, C_{2N-1}\}$ is given to the Nth switch block, a single $(4N+1)$-stage RO is formed instead of the two separate loops in the BR and the RO. To avoid this configuration, we assign the value of $\mathrm{XOR}\{C_1, \cdots, C_{2N-1}\}$ to the switch block at the $2N$th stage. Hence, the number of effective challenge bits for the CF-PUF becomes $2N-1$.

B. Nonlinearity of the Response

As discussed above, CF-PUF is designed to determine a response based on the convergence time dominantly according to a nonlinear function of the threshold voltage. Let the oscillation period of the RO and the convergence time of the BR be T_{RO} and T_{BR}, respectively. The response of CF-PUF, R_{CF}, can be expressed as

$$R_{\mathrm{CF}} = \left\lfloor \frac{T_{\mathrm{BR}}}{T_{\mathrm{RO}}/2} \right\rfloor \bmod 2. \tag{3}$$

The response contains two nonlinear components: T_{BR}, which is a nonlinear function of the threshold voltage, and the modulo operation, which is a nonlinear function itself. Thus, CF-PUF is considered resistant to ML attacks.

CF-PUF also possesses stronger immunity than RO-PUF against side-channel attacks as CF-PUF contains a BR structure [10]; RO-PUF is vulnerable against electromagnetic analysis for detecting the frequencies of the RO, which can then be used to mathematically clone the PUF. However, the frequency of the BR changes temporally as the number of pulse edges changes, so the responses of CF-PUF are difficult to retrieve.

Possible side-channel attacks specific to CF-PUF include current measurement, which indirectly reveal the

TABLE I
UNIQUENESS EVALUATIONS FOR BR-PUF AND CF-PUF

	H	D	U
16-bit BR-PUF	0.25	0.53	0.79
18-bit BR-PUF	0.26	0.54	0.87
17-bit CF-PUF	**0.98**	**1.00**	**0.89**

oscillation duration of the BR. However, the aforementioned nonlinear properties make it difficult to model the relationship between the challenge and the oscillation duration. Though the results are omitted due to space limitation, SPICE-based experiments that simulate attempts to predict the T_{BR} value using regression analyses were all unsuccessful (the results are not shown here due to space limitations).

V. NUMERICAL EVALUATION

The performances of the proposed 17-bit CF-PUF and two conventional 16-/18-bit BR-PUFs (long and normal) are evaluated using SPICE simulations. Since CF-PUF and BR-PUF can only take odd and even numbers of challenges, respectively, the above 3 conditions with similar numbers of challenge bits were selected to conduct a fair comparison.

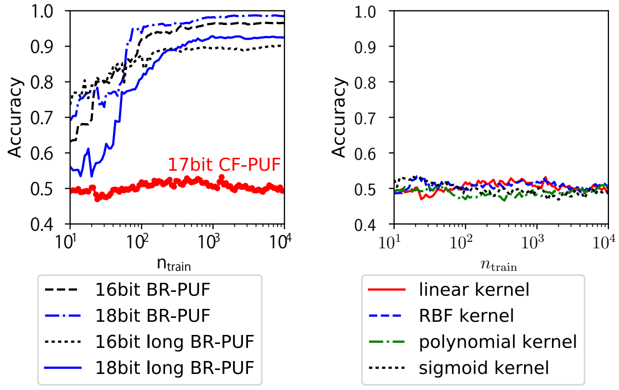
The commercial SPICE simulator and a 65 nm process library and the same threshold voltage variation described in the previous section were used. For each PUF, 12,000 CRPs obtained using random challenges were evaluated in terms of 5 criteria: randomness, diffuseness, uniqueness, resistance against ML attacks, and reliability. The simulation period was 1.0×10^{-7} s, within which all of the BR-PUFs and all but one of CF-PUFs converged.

A. Randomness, Diffuseness, and Uniqueness

We first evaluate the performances of the PUFs in terms of the following 3 metrics: randomness (H), diffuseness (D), and uniqueness (U) [5]. Randomness represents the similarity of the appearances of 0's and 1's in a PUF's response. Diffuseness represents whether a single PUF returns different responses to different challenges, and uniqueness represents whether different PUFs return different responses for the same challenge. Each metric takes a value between 0 and 1, where 0 is the worst and 1 is the best. A total of 1000 CRPs were generated for 10 different PUF instances. The results shown in Table I indicate that CF-PUF outperforms the BR-PUFs in terms of all 3 of these metrics.

B. Resistance against Machine-Learning Attacks

The resistance against ML attacks is a characteristic of a PUF representing the difficulty to predict its CRPs using ML techniques. This characteristic was evaluated in terms of the prediction accuracy by an SVM classifier with a linear kernel implemented by a Python scikit-learn library [13]. Two thousand CRPs were randomly selected for testing, and n_{train} CRPs were selected without overlap for training. The prediction accuracy was calculated with different n_{train} values. The ideal prediction accuracy is 0.5, which indicates that the response is completely unpredictable.



(a) With linear kernel.

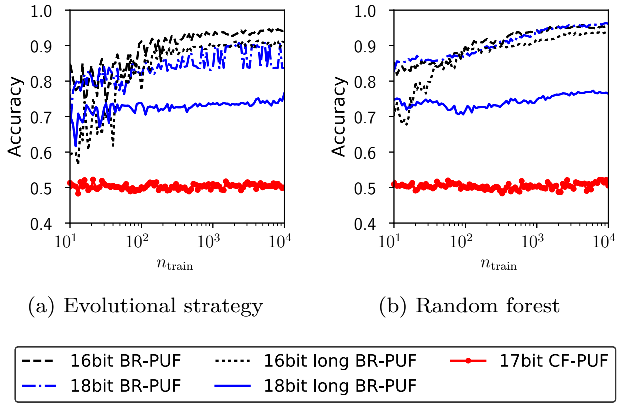
(b) 17bit CF-PUF against various kernels.

Fig. 7. Prediction accuracy by SVM attacks.

The simulation results are shown in Fig. 7(a). In this experiment, in addition to the BR-PUF and CF-PUF, 12,000 CRPs with convergence times that are longer than 2.5×10^{-8} s were also evaluated and referred to "long BR-PUF." In the 16-bit and 18-bit BR-PUFs, the prediction accuracy was above 0.9, indicating that their responses are generally predictable by the SVM after about 100 CRPs are collected for training. The prediction accuracy in the long BR-PUF was lower than that of the BR-PUF, but still close to 0.9. On the other hand, the prediction accuracy in CF-PUF remains constant at about 0.5 regardless of the number of CRPs collected.

In addition, the prediction accuracy of CF-PUF by an SVM classifier using a radial basis function (RBF) kernel, a polynomial kernel, and a sigmoid kernel was tested. As shown in Fig. 7(b), the prediction accuracy in CF-PUF remained around 0.5, indicating that it is difficult to predict the response of CF-PUF by these various SVM-based ML attacks even when more than 10,000 training samples are used.

We also evaluated the resistance of the proposed PUF against 4 ensemble ML techniques that are known to be effective attacks against PUFs: the evolution strategy (ES) [14], random forest (RF) [15], bagging [16], and boosting [17]. ES was implemented by Python with PyBrain [18], and the other classifiers were implemented with scikit-learn [13]. The results of the first 2 attacks are shown in Fig. 8; the results of the last 2 are omitted due to space limitations, but the results were very similar to those shown in Fig. 8. The results confirm that the proposed CF-PUF was also resistant to ensemble ML methods. Together, these results demonstrate that the proposed CF-PUF has strong resistance against a wide variety of ML attacks.



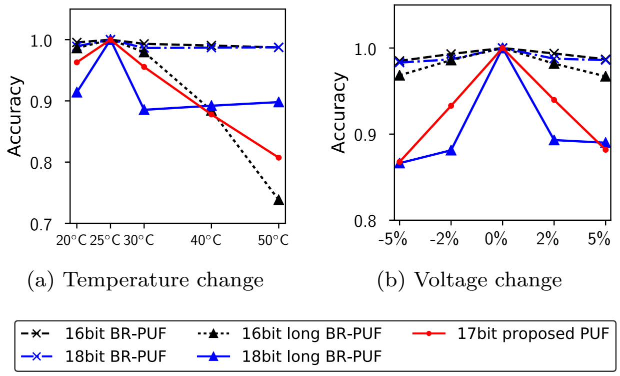
(a) Evolutional strategy

(b) Random forest

Fig. 8. Prediction accuracy by ensemble ML techniques.

C. Reliability

Reliability is a characteristic whereby the same PUF always returns the same response to the same challenge. Here, the reliability of CF-PUF and various BR-PUFs was evaluated as the temperature and the supply voltage were varied. The PUFs were evaluated at measurement temperatures of 20°C, 30°C, 40°C, and 50°C, and the results were compared to that at 25°C as a reference. While the supply voltage was varied from the nominal voltage of 1.2 V by −2%, +2%, −5%, and +5%. The results in Fig. 9 show that BR-PUF is slightly better than CF-PUF and long BR-PUFs under both temperature changes and variations in the supply voltage. However, considering the results shown in Sec. V-B, this reduced reliability of CF-PUF compared to the BR-PUF (only down to 0.9) comes with an improved resistance against ML attacks. Thus, under appropriate voltage and temperature control, CF-PUF is better than the BR-PUF.



(a) Temperature change

(b) Voltage change

Fig. 9. Reproducibility under temperature and voltage changes.

VI. Conclusion

In this paper, we proposed a novel CF-PUF that utilizes the convergence time of a BR such that the response of the CF-PUF is generated based on a nonlinear function. This feature makes it difficult to predict the CRPs in an ML attack using a linear classifier. Through the SPICE simulations, it was shown that the use of ML techniques to predict the responses of a CF-PUF was only about 0.5, which means that it is highly unpredictable.

References

[1] B. Gassend, D. Clarke, M. van Dijky, and S. Devadas, “Silicon physical random functions,” in *Proceedings of Computer and Communication Security Conference*, Nov. 2002, pp. 148–160.

[2] G. E. Suh and S. Devadas, “Physical unclonable functions for device authentication and secret key generation,” in *Proceedings of Design Automation Conference*, Jun. 2007, pp. 9–14.

[3] J. Guajardo, S. Kumar, G. Schrijen, and P. Tuyls, “FPGA intrinsic PUFs and their use for IP protection,” *Proceedings of Cryptographic Hardware and Embedded Systems*, pp. 63–80, 2007.

[4] Q. Chen, P. L. G. Csaba, U. Schlichtmann, and U. Rührmair, “The bistable ring PUF: A new architecture for strong physical unclonable functions,” in *Proceedings of IEEE International Symposium on Hardware-Oriented Security and Trust*, 2011, pp. 134–141.

[5] Y. Hori, T. Yoshida, T. Katashita, and A. Satoh, “Quantitative and statistical performance evaluation of arbiter physical unclonable functions on FPGAs,” in *Proceedings of International Conference on Reconfigurable Computing*, Dec. 2010, pp. 298–303.

[6] U. Rührmair, J. Sölter, F. Sehnke, X. Xu, A. Mahmoud, V. Stoyanova, G. Dror, J. Sölter, F. Sehnke, and S. Devadas, “PUF modeling attacks on simulated and silicon data,” *IEEE Transaction on Information Forensics and Security*, vol. 8, pp. 1876–1891, Nov. 2013.

[7] G. Hospodar, R. Maes, and I. Verbauwhede, “Machine learning attacks on 65nm arbiter PUFs: Accurate modeling poses strict bounds on usability,” in *Proceedings of IEEE Workshop on Information Forensics and Security*, Dec. 2012, pp. 37–42.

[8] X. Xu, U. Rührmair, D. E. Holcomb, and W. Burleson, “Security evaluation and enhancement of bistable ring PUFs,” in *Proceedings of International Workshop on Radio Frequency Identification*, 2015, pp. 3–16.

[9] A. Winstanley and M. Greenstreet, “Temporal properties of self-timed rings,” in *Proceedings of Advanced Research Working Conference on Correct Hardware Design and Verification Methods*, 2001, pp. 140–154.

[10] A. Cherkaoui, L. Bossuet, and C. Marchand, “Design, evaluation, and optimization of physical unclonable functions based on transient effect ring oscillators,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1291–1305, June 2016.

[11] K. Kanda, K. Nose, H. Kawaguchi, and T. Sakurai, “Design impact of positive temperature dependence on drain current in sub-1-V CMOS VLSIs,” *IEEE Journal of Solid-State Circuits*, vol. 36, no. 10, pp. 1559–1564, Oct 2001.

[12] *HSPICE User Guide: Basic Simulation and Analysis Version L-2011.09*, Synopsys, Inc., 2011.

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel *et al.*, “Scikit-learn: Machine learning in python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[14] H.-G. Beyer and H.-P. Schwefel, “Evolution strategies – a comprehensive introduction,” *Natural Computing*, vol. 1, no. 1, pp. 3–52, 2002.

[15] V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, and B. P. Feuston, “Random forest: A classification and regression tool for compound classification and qsar modeling,” *Journal of Chemical Information and Computer Sciences*, vol. 43, no. 6, pp. 1947–1958, 2003.

[16] L. Breiman, “Bagging predictors,” *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[17] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119 – 139, 1997.

[18] T. Schaul, J. Bayer, D. Wierstra, Y. Sun, M. Felder, F. Sehnke, T. Rückstieß, and J. Schmidhuber, “PyBrain,” *Journal of Machine Learning Research*, vol. 11, pp. 743–746, 2010.