A PUF-Based Paradigm for IoT Security

Tarek Idriss, Haytham Idriss, Magdy Bayoumi, *Fellow IEEE*
The Center for Advanced Computer Studies
University of Louisiana at Lafayette
Lafayette, Louisiana, USA

***Abstract*—Physically Unclonable Functions or PUFs have emerged as a promising lightweight authentication alternative to traditional cryptography. This made them a promising candidate for constrained IoT devices. These PUFs however have been found to be vulnerable to modeling attacks to varying degrees. The security of different PUFs designs against these modeling attacks have been evaluated in multiple research efforts. However, a more accurate evaluation that incorporates the area footprints of the different PUFs is missing from the literature. Such evaluation is rather essential in order to determine the most suitable PUF design for the different constrained IoT devices. To perform this evaluation, we implemented several strong delay-based PUFs. The results showed that when factoring in the PUFs implementation area costs, these PUFs compare differently as some enhanced PUF designs turned out to have inferior efficiency compared to their simpler counterpart. We recognized the most effective design elements in the implemented PUFs. Based on the efficiency figures obtained, we recommended the optimal PUF design for the different security schemes used in constrained IoT devices.**

***Index Terms*—PUF, lightweight authentication, modeling attacks, IoT, hardware security, constrained devices.**

I. Introduction

Billions of devices will compromise the internet of things (IoT). It is expected that more than 50 billion devices will be connected to the internet by 2020 [1]. The security of these devices is one of the major concerns for the wide adaptation of IoT [2]. The IoT naturally inherits all the security concerns of the infrastructure it utilizes which includes various technologies like cloud computing servers, wireless sensor nodes and even RFIDs. Low power device security, however, remains a particular concern for IoT as many of the newly incorporated devices will be critically powered. Such devices can be battery-less like passive RFID tags which have strict area and power limitations. Other critically powered devices are battery operated devices that are infeasible to recharge or supply with new batteries. Examples of these devices include wireless sensor nodes or medical implants. The security of the authentication process in some of these small devices is of critical importance. However, standard cryptographic solutions that provide provable security have prohibitive area and power demands for such devices. For instance, low cost RFID tags can only use 3-5K logic gates for security functions as reported in [3] and [4] while public cryptography algorithms implementations which are crucial for reliable authentication can use between 12 to 22K logic gates [5].

Silicon Physical Unclonable Functions (PUFs) have been proposed as a low cost alternative to standard authentication protocols [6]. These PUFs exploit the randomness inherent within the manufacturing process of CMOS devices to produce unique responses. Of the many suggested PUFs, delay based arbiter PUFs [6] and their variations [7], [8] and [9] are of the most studied and understood PUFs. Several authentication protocols that utilize such PUFs have been suggested in [10], [11] and [12]. These PUFs were, however, found to be vulnerable to modeling attacks as shown in [13], [14] and [15]. In such attacks, an adversary would collect the exchanged challenge and response pairs (CRPs) used in authentication and apply machine learning algorithms in order to produce a software model of the PUF that is capable of correctly predicting the response of new challenges.

The authors in [13] have performed an extensive analysis on the effectiveness of modeling attacks against popular PUF designs. They reported and described the security for each PUF design using two values: the number of CRPs that needs to be collected and the time required to model a given PUF. Tobisch, J. and Becker, G. T. confirmed part of the work made in [13] as they reported similar values in [14]. While the reported security values in [13] and [14] showed how different delay based PUFs offer different levels of security against modeling attacks, there has been no work in literature that shows how these PUFs compare in resisting modeling attacks when factoring their implementation area footprint. In order to compare these PUFs in term of their efficiency in producing the desired security, we have implemented each in 45nm CMOS technology and then compared their security values while factoring in the design area footprint. The results showed that when factoring in the PUF area, these PUFs compare differently. We were thus able to recognize the most effective design elements in the implemented PUFs. We then classified the IoT PUF security schemes and recommended the optimal design for each based on the efficiency figures produced in our work. This paper is divided as follows: Section II is an overview of PUFs. Section III describes the architecture of the implemented PUFs. In section IV we classified the PUF security schemes according to the level of CRP accessibility and suggested the most suitable IoT devices for each. In Section V we report and compare the efficiency values for the implemented PUFs. In Section VII we discuss the implication of our results and we suggest the optimal PUF designs for each security scheme. We conclude in Section VIII.

978-1-5090-4130-5/16/$31.00 ©2016 IEEE

II. PUF Overview

A PUF is a physically disordered system that reacts to an external challenge with a unique unpredictable response [16]. Although the response cannot be predicted from the circuit, it is reliable, i.e. a challenge ‘C’ will, ideally, always produce the same response ‘R’. Hence the PUF design and layout are not a secret and their unclonability relies on the miniscule variations during fabrication.

The challenge-response space of a PUF determines whether it is classified as a weak PUF or a strong PUF [16]. Strong PUFs have a large challenge-response space that grows exponentially with its implementation size, this allows their usage in challenge/response authentication protocols where each challenge response pair is used only once.

Weak PUFs are characterized by having a small challenge-response space with respect to their implementation size. Such PUFs are used as an alternative to secure key storage. By having a hidden response, these PUFs will be generating an agreed upon key reliably as alternative to saving the key in non-volatile memory which is highly vulnerable to extraction.

III. Delay Based PUFs

In this section, we familiarize the reader with the design and architecture of the delay-based strong PUFs examined: The Arbiter PUF [6], the XOR-Arbiter PUF [7], the Lightweight Secure PUFs [8] and the Feed Forward (FF) Arbiter PUF [9].

A. Arbiter PUF

Arbiter PUFs [6] compare the delay of two identical paths to generate either a ‘0’ or a ‘1’ bit depending on the result of the comparison. Although the two paths are identical and should introduce the same delay, unpredictable miniscule variations during the fabrication process ensures that one path is ultimately faster than the other.

Multiplexers, referred to as “Switch Components”, are inserted into the paths. Challenge bits are used as select inputs to the multiplexers. Each switch component either flips the compared paths or keeps them the same. This result in a high combination of possible paths. Figure 1 shows the architecture of such PUF.

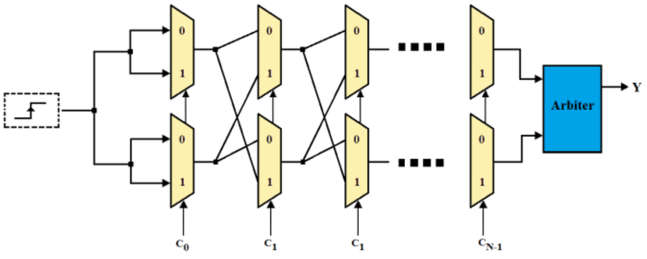

Fig. 1. Arbiter PUF architecture

B. XOR-Arbiter PUF

The XOR-Arbiter PUF [7] or XOR PUF combines several rows of the simple arbiter PUFs by XORing their output into one single bit. This makes the circuit more resistant to modeling attacks. Two design parameters describe an XOR-Arbiter PUF: the challenge bit-length and the input size of the XOR which indicates the number of rows utilized.

C. Lightweight Secure PUF

Majzoobi, M., Farinaz K., and Miodrag P introduced the Lightweight Secure PUF (LS PUF) in [8]. Their design introduces an input network that rearranges and alters the way the challenge inputs are fed to ensure increased hamming distance for the challenge inputs. An output network is also introduced to XOR several arbiter values together while using the same arbiters in multiple XORs to increase the throughput of the design. For an in-depth description, the reader can refer to the authors’ original work in [8].

In our implementation, we use only one XOR gate as the output network. This is to keep in accordance with the security values in [13] that utilized such design. Note that this represent an upper bound on the security of the Lightweight secure PUF, as using more XOR gates on the same rows would further expose the circuit.

D. Feed Forward PUFs

Feed Forward PUFs or FF PUFs introduce non-linearity by utilizing feed forward paths that are inserted in the circuit [9] as shown in Fig. 2. Two parameters govern the design of the FF PUFs: The challenge bit length and the number of feedback loops.

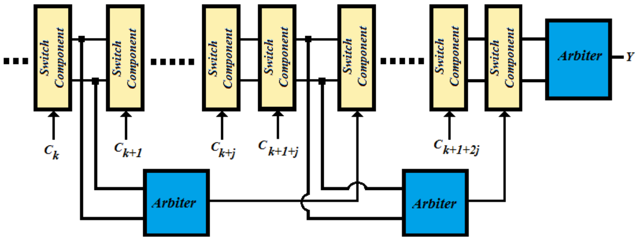

Fig. 2. FF PUF Architecture

E. Resistance to Modeling Attacks

These delay based arbiter PUFs and their variants were shown to be vulnerable to modeling attacks in [13], [14] and [15]. Such attacks utilize machine learning methods to correctly predict the output of new challenges by analyzing extracted CRPs. These CRPs can be extracted by physically probing the circuit itself, by continuously listening and recording authentications messages received and sent by the device, or by impersonating a legitimate authenticator and polling the device repetitively.

The security of a given PUF, in regards to resisting modeling attacks, can be parameterized by two values [13]:

1) NCR: the number of CRPs that needs to be collected in order to successfully model the PUF with 99% predictability.

2) *TTM:* the time required to successfully model a PUF given that enough CRPs are collected.

For the PUF to be secure it has to either resist modeling attacks by requiring a large number of CRPs that is infeasible to collect (a very large NCR) or by requiring an infeasible amount of time to produce a model of the PUF (a very large TTM).

IV. PUF Security Schemes in IoT Applications

In this section, we classify the PUF security schemes based on an adversary's ability to collect exchanged CRPs. We recognize three different accessibility levels for the CRPs: hidden, limited and fully accessible. We mention the possible applications and IoT devices that might employ each.

A. CRP Hidden

In such schemes, the transmitted challenges and responses are completely hidden from an adversary. This is the case for many Controlled PUFs [17] (CPUFs) where the transmitted challenge and response are transformed through cryptographic functions or pseudo-cryptographic functions. In these schemes, the device should also be secured against probing attacks. This can be either done be denying physical access or by utilizing probe-resistant technology. For reliable security, it is essential to use proven cryptographic functions at the cost of increased area requirement. Such schemes are best suited for personal low power mobile devices that are under their owner supervision. These can include wearable computing/medical devices that have essential security needs and are critically powered. For these devices, a PUF's NCR and TTM values are rather irrelevant as we assume it's impossible to model the PUF due to the lack of access to the CRPs. As such, we recommend PUFs with lowest area implementation requirement as the control logic used to hide the CRP exchanges will add a significant area overhead.

B. CRP Limited

In this scheme, the CRPs can be collected in a limited fashion. The attacker might only be able to collect the CRPs transmitted and requested by trusted parties. Another possibility is having the PUF polling rate artificially slowed to ensure that an adversary would not be able to poll the device at a sufficient speed. Probing attacks would still be a problem and the device should be secured against them in a similar fashion to the PUFs used in the CRP Hidden scheme. This PUF security scheme does not require cryptographic control functions and as such it is suitable for devices with stricter area/power limitations. These devices can be ultra-low power devices such as passive RFID tags used for personal identification like in passports or ID cards. The security of PUFs implemented in such schemes would be entirely dependent on the achieved NCR value.

TABLE I. PUF Security Schemes Classification

CRP Accessibility	***Suggested devices***	***Restriction on Area/Power***
Hidden	Low power wearable computing or medical devices	High restriction
Limited	Passive RFID tags in ID cards or passports	Very high restriction
Full	Deployed wireless sensor networks	Low/medium restriction

C. CRP Fully Accessible

In this scenario the required number of CRPs can be collected by an adversary either by polling the device at a high speed or by simply probing the device. Devices that might utilize these PUFs are unmanned devices such as wireless sensor nodes deployed for infrastructure monitoring, environmental monitoring or espionage missions. PUFs used in such attacks should have sufficient TTM security to prevent adversaries from modeling the PUF in a feasible amount of time. Table I summarizes the security classification we proposed.

V. Implementation Results and Comparison

To identify the efficiency of the PUF designs in resisting modeling attacks, each design has been implemented in 45nm CMOS technology using the FreePDK45 digital library [18]. Synthesis results were generated by Cadence Encounter. The area requirement of each design is reported in gate equivalent (GE) unit. The NCR and TTM values of each PUF were collected from [13] and [14]. Both of these values are then normalized according to the area requirement of the corresponding PUF design. This allowed us to produce their respective security efficiency:

a) NCR/GE: "Achieved NCR area efficiency". Indicates the PUF's ratio of achieved NCR over its area implementation cost.

b) TTM/GE: "Achieved TTM area efficiency". Indicates the PUF's ratio of achieved TTM over its area implementation cost.

It is noted that the security figures reported in [13] and [14] are not hard figures as they are based on experimental results. Nevertheless, these figures are accurate enough for comparing the security of each PUF particularly when the difference is significant. The result of the PUF implementations is inspected and the security efficiency of each is evaluated.

A. Arbiter PUF

The NCR for the Arbiter PUF were obtained from [13]. The required implementation areas for all PUFs were obtained from our implementations. Table II shows the NCR Area efficiency of the Arbiter PUF for different challenge bit-lengths. The area efficiency *NCR/GE* of the Arbiter PUF is constant as the NCR growth is linear.

TABLE II. Arbiter PUF NCR Efficiency Growth

Challenge Length	***64-bit***	***128-bit***	***256-bit***
NCR	3300	6500	12900
Area (GE)	351	698	1394
Efficiency (NCR/GE)	9.395	9.312	9.250

The TTM values reported in [13] show polynomial growth for its efficiency. We compare the Arbiter PUF TTM efficiency with the rest of the implemented PUFs in Fig. 6.

B. XOR Arbiter PUF

Figure 3 shows the NCR area efficiency of both 64-bit and 128-bit implementations of the XOR PUF. The NCR values were obtained from [14]. These values were also in accordance with the work performed in [13]. It is noted that the area efficiency increases as we increase the number of XORed rows in the design. Figure 3 also shows that the 64-bit implementations of the XOR PUFs are more efficient than their 128-bit counterparts. For instance, a 64-bit, 8-input XOR PUF offers a much higher NCR value when compared to a 128-bit 4-input XOR PUF while consuming the same area on chip.

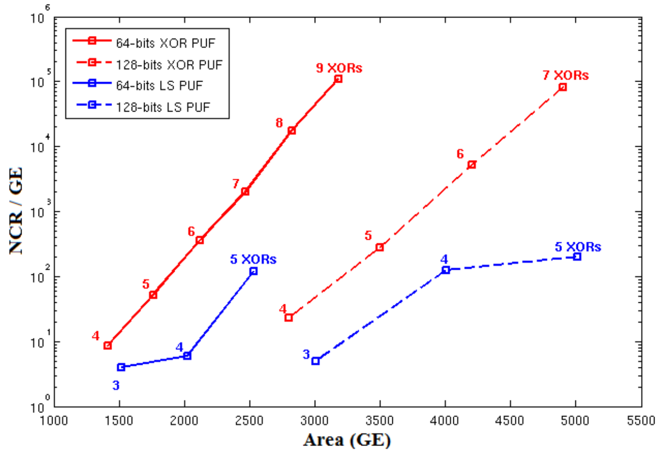

Fig. 3. NCR Area Efficiency Growth for XOR PUF and LS PUF

Figure 4 shows the TTM area efficiency growth of the implemented XOR PUFs. The TTM values were obtained from [14]. Similar to their NCR area efficiency growth, the TTM area efficiency also increases as we increase the number of XORed rows. Again, the 64-bit implementations show a much higher efficiency when compared to their 128-bit counterpart.

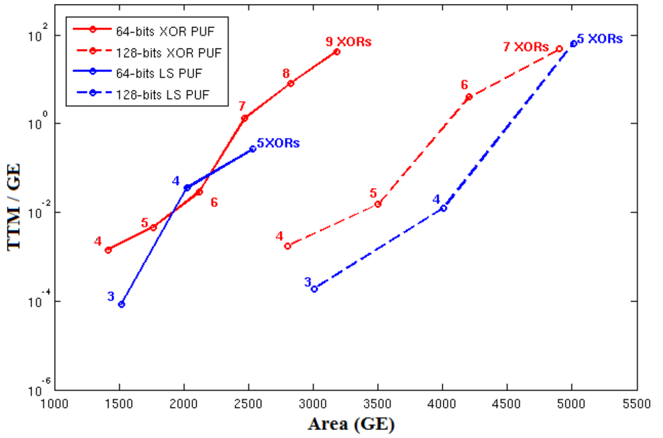

Fig. 4. TTM Efficiency Growth for XOR PUF and LS PUF

C. LS Arbiter PUF

Figure 3 shows the NCR area efficiency of the 64-bit and 128-bit implementations of the LS PUF. The NCR values were obtained from [13]. Similar to the XOR-Arbiter, the area efficiency increases as we increase the number of XORed rows with the 64-bit implementations being more efficient than the 128-bit implementations.

Figure 4 shows the TTM area efficiency growth of the LS PUF. The TTM values were also obtained from [13]. The efficiency increases with the number of XORed rows while favoring the 64-bit implementations.

D. FF Arbiter PUF

Table III shows the NCR area efficiency of the FF PUF with different challenge bit-lengths. NCR values were estimated based on the equations conjectured in [13]. The efficiency is almost constant as the equation suggests a linear growth with respect to the number of delay elements in the design.

TABLE III. 3-Loops FF PUF NCR Efficiency Growth

Challenge Length	***64-bit***	***128-bit***	***256-bit***
NCR	19353	37567	73997
Area (GE)	380	726	1423
Efficiency (NCR/GE)	50.92	51.74	52

We were not able to obtain values for the TTM area efficiency of the FF Arbiter PUF as there is a lack of such values in literature. However, Ruhmair et al in [13] suggested a polynomial growth of slightly higher order than that of the simple Arbiter PUF.

E. PUF Efficiency Comparison

Figure 5 shows a comparison between the implemented PUFs in terms of their NCR area efficiency. Both the arbiter PUF and the FF PUF are orders of magnitude behind the XOR and the LS PUF in terms of their security efficiency which rather remain constant. This is due to the linear growth of their security when compared to the exponential growth of the XOR and LS PUFs. The FF Arbiter PUF does, however, offer a better NCR area efficiency when compared to the normal Arbiter PUF and has the best NCR area efficiency of all implemented PUFs for area restriction below 1765 GE.

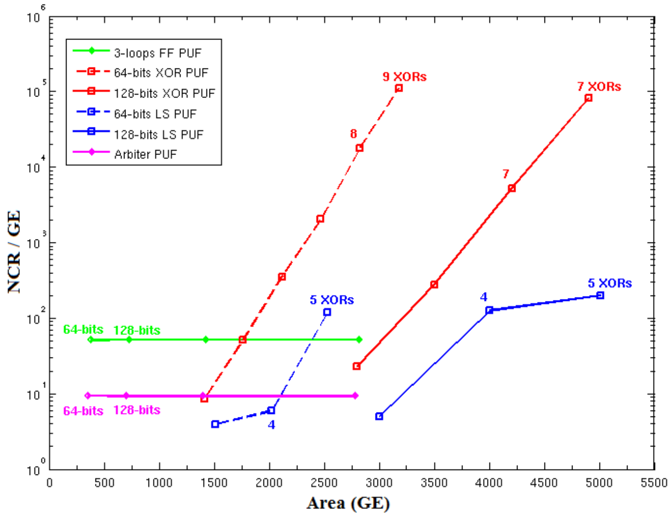

Fig. 5. NCR Efficiency Growth for All Implemented PUFs

Comparing the NCR area efficiency of the XOR PUF with the LS PUF in Fig. 5, we find that the XOR PUF is more efficient. Although the LS PUF required the collection of more CRPs according to [13], when comparing it to the XOR PUF in terms of its area efficiency the XOR PUF is the clear winner. This is a rather important result as the LS PUF was introduced as an improvement over the XOR PUF. The area overhead imposed by its input network (calculated to be around 43% in our implementation) makes it much less efficient in regards to its NCR security. This additional area could be used to implement a larger XOR PUF capable of producing a higher NCR value.

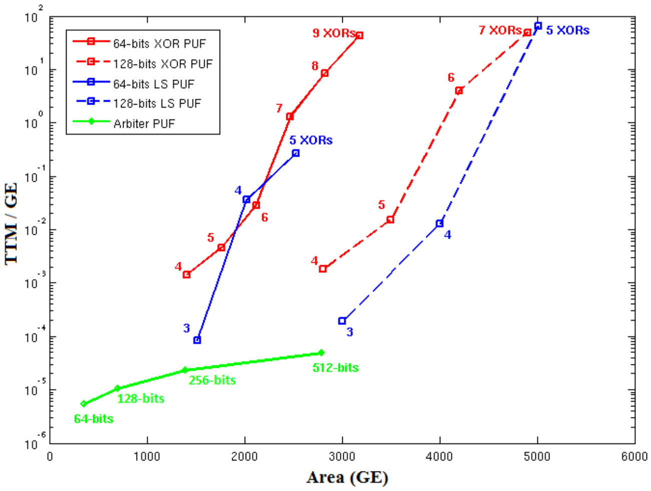

Fig. 6. TTM Area Efficiency Growth for Implemented PUFs

Figure 6 shows a comparison between the TTM area efficiency of the XOR PUF and the LS PUF along with the Arbiter PUF. The XOR PUF surprisingly seems to have slightly better TTM efficiencies in comparison to the LS PUF. The difference, however, is rather marginal considering that the values we obtained are based on the rather limited number of experiments performed in [13]. Still, such result was unexpected as the work in [13] reported a significant increase in the number of trials needed by the machine learning algorithm when modeling the LS PUF.

Figure 6 also shows how the Arbiter PUF severely lags behind the XOR PUF and LS PUF in terms of its TTM area efficiency.

VI. PUF Design Implications and Use in Devices

The security efficiency values obtained through our work can serve as a guideline for PUF design. The efficiency values in Fig. 5 and Fig. 6 shows that utilizing XOR output networks provide the most effective increase in security as opposed to utilizing XOR input networks in LS-PUF. The results also show that feed forward loops are very efficient in providing increased security. The FF PUF which utilized 3 FF loops provided significant increase in NCR security compared to the traditional arbiter with minimal area overhead. New PUF designs should incorporate the recognized effective elements and avoid the ones with minimal impact.

The security efficiency would also allow us to recognize the most suitable PUF to use for the different security schemes identified earlier. Table 4 summarizes the recommended choice of PUF to use in the different security schemes.

For the hidden CRP PUF security schemes, the PUFs with the lowest area requirements are the most suitable as the security values are irrelevant when no CRPs can be collected by an adversary. The FF PUF and the Arbiter PUF both have similar area implementation costs albeit the FF PUF provides slightly better security. The FF PUF added security would have very little impact on the security of the system overall as the latter depends heavily on the cryptographic functions utilized to hide the CRPs. For this reason, we recommend the use of the Arbiter PUF which have the smallest area requirement.

TABLE IV. PUF Design Schemes Requirement and Recommendation

	CRP Accessibility		
	Hidden	***Limited***	***Full***
PUF Design Criteria	Minimum Area	Highest NCR Efficiency	Highest TTM Efficiency
Recommended PUF	Arbiter PUF	XOR PUF (>1765 GE) FF PUF (<1765 GE)	LS-PUF / XOR-PUF

For PUF security schemes with limited CRPs accessibility, the most suitable PUFs are the ones the highest NCR area efficiencies. Such PUFs would produce the largest number of CRPs required to model the PUF. Inspecting our implementation results in Fig. 5, we find the XOR PUF to be the most efficient design when it comes to its NCR values. It is noted though that at very strict area constraints (<1765 GE), the FF PUF will be the most efficient in terms of its NCR security.

PUF security schemes that utilize accessible CRPs architecture will depend on achieving an extremely high TTM value that would practically prevent any kind of modeling attack. Both the LS PUF and the XOR PUF have similar TTM area efficiency. Nevertheless, in order to achieve the required TTM values, very large PUFs need to be implemented. In [13] an XOR PUF with 512-bit challenge length and 8-input XORs was found to be immune against modeling attack. According to our implementation, such PUF would require 22,301 GE. A similar LS PUF would require 31,931 GE. Both implementations cannot be employed by RFID tags which can only use up to 5K logic gates for security. Such large PUFs are also well beyond what is considered suitable for “lightweight” IoT devices and as such it might be more reasonable to use standard security for such exposed devices.

VII. Conclusion

In this paper, we have classified the PUF security schemes according to the exposure of the transmitted CRPs. We suggested suitable IoT devices for each scheme and established the most important PUF design criteria for each. While prior work evaluated the security of different PUF designs, the area footprint overhead was never factored in. Such overhead is important to consider when the intended PUF usage is meant for constrained devices. In order to evaluate the area efficiency of these PUFs, we implemented each in 45nm technology. Our work showed that when factoring in the PUF area, these PUFs compare differently. Such area efficiency evaluation is necessary as PUFs are intended for constrained devices. The LS PUF proved to have lower threshold of required CRPs for modeling (NCR security) than prior simpler PUFs (XOR-PUF) when its area overhead is factored in. The FF PUF showed the best NCR security value for highly restricted implementation areas (<1765 GEs). Based on the efficiency figures obtained, we were able to recognize the most effective design elements in the implemented PUFs to be further employed by new PUF designs. We also recommended the optimal PUF design for each security scheme.

References

[1] M. A. Feki, F. Kawsar, M. Boussard and L. Trappeniers, "The Internet of Things: The Next Technological Revolution," in Computer, vol. 46, no. 2, pp. 24-25, Feb. 2013.

[2] Zhao, Kai, and Lina Ge. "A survey on the internet of things security." *Computational Intelligence and Security (CIS), 2013 9th International Conference on*. IEEE, 2013.

[3] Chien, Hung-Yu. "SASI: A new ultralightweight RFID authentication protocol providing strong authentication and strong integrity." IEEE Transactions on Dependable and Secure Computing 4.4 (2007): 337-340.

[4] Sarma SE. “Towards the five-cent tag.” Technical ReportMIT-AUTOID-WH-006, MIT Auto ID Center, 2001.Available at: http://www.autoidlabs.org [2001].

[5] Liu, Dongsheng, et al. "Design and Implementation of An ECC-Based Digital Baseband Controller for RFID Tag Chip." IEEE Transactions on Industrial Electronics 62.7 (2015): 4365-4373.

[6] Gassend, B., et al., "Silicon physical random functions." Proceedings of the 9th ACM conference on Computer and communications security. ACM, 2002.

[7] Suh, G. E., & Srinivas D., "Physical unclonable functions for device authentication and secret key generation." *Proceedings of the 44th annual Design Automation Conference*. ACM, 2007.

[8] Majzoobi, M., Farinaz K., & Miodrag P., "Lightweight secure pufs." *Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design*. IEEE Press, 2008.

[9] Lee, J. W., et al., "A technique to build a secret key in integrated circuits for identification and authentication applications." *VLSI Circuits, 2004. Digest of Technical Papers. 2004 Symposium on*. IEEE, 2004.

[10] Devadas, S., et al., "Design and implementation of PUF-based" unclonable" RFID ICs for anti-counterfeiting and security applications." *2008 IEEE International Conference on RFID*. IEEE, 2008.

[11] Pappu, R., Recht, B., Taylor, J., & Gershenfeld, N. (2002). Physical one-way functions. *Science*, *297*(5589), 2026-2030.

[12] Rostami, M., Majzoobi, M., Koushanfar, F., Wallach, D. S., & Devadas, S. (2014), "Robust and reverse-engineering resilient PUF authentication and key-exchange by substring matching." *IEEE Transactions on Emerging Topics in Computing* 2.1 (2014): 37-49.

[13] Rührmair, U., et al., "PUF modeling attacks on simulated and silicon data." *IEEE Transactions on Information Forensics and Security* 8.11 (2013): 1876-1891.

[14] Tobisch, J., & Georg T. B., "On the Scaling of Machine Learning Attacks on PUFs with Application to Noise Bifurcation." *International Workshop on Radio Frequency Identification: Security and Privacy Issues*. Springer International Publishing, 2015.

[15] Ganji, F., Shahin T., & Seifert, J.P., "Why attackers win: on the learnability of XOR arbiter PUFs." *International Conference on Trust and Trustworthy Computing*. Springer International Publishing, 2015.

[16] Guajardo, J., et al. "FPGA intrinsic PUFs and their use for IP protection." *International workshop on Cryptographic Hardware and Embedded Systems*. Springer Berlin Heidelberg, 2007.

[17] Gassend, B., Clarke, D., Van Dijk, M., & Devadas, S. (2002). "Controlled Physical Random Functions." *Security with Noisy Data*. Springer London, 2007. 235-253.

[18] Stine, James E., et al. "FreePDK: An open-source variation-aware design kit." *2007 IEEE International Conference on Microelectronic Systems Education (MSE'07)*. IEEE, 2007.