Quo Vadis, PUF?

Trends and Challenges of Emerging Physical-Disorder based Security

Masoud Rostami*, James B. Wendt†, Miodrag Potkonjak†, and Farinaz Koushanfar*
* Department of Electrical and Computer Engineering, Rice University, Houston, TX.
Email:{masoud, fariaz}@rice.edu
† Computer Science Department, University of California, Los Angeles, CA.
Email: {jwendt, miodrag}@cs.ucla.edu

***Abstract*—The physical unclonable function (PUF) has emerged as a popular and widely studied security primitive based on the randomness of the underlying physical medium. To date, most of the research emphasis has been placed on finding new ways to measure randomness, hardware realization and analysis of a few initially proposed structures, and conventional secret-key based protocols. In this work, we present our subjective analysis of the emerging and future trends in this area that aim to change the scope, widen the application domain, and make a lasting impact. We emphasize on the development of new PUF-based primitives and paradigms, robust protocols, public-key protocols, digital PUFs, new technologies, implementations, metrics and tests for evaluation/validation, as well as relevant attacks and countermeasures.**

I. Introduction

Modern security has many faces and must cover a wide spectrum of tasks. In addition to classical cryptography that provides security of stored or communicated data, modern security has to address a variety of other requirements including trust, anonymity, and privacy of actions. Earlier cryptographic methods and protocols mostly aimed to provide security for physically well-protected devices. However, a majority of contemporary devices (e.g., RFIDs or nodes in sensor networks) are easily accessible and physically unprotected, while they may even reside in hostile environments. Therefore, modern security primitives and protocols must be resilient to physical and side channel attacks. They must also be inexpensive and low power to fulfill the constraints of portable computing and communicating devices.

To address these shortcomings, an alternative security approach based on the inherent, unclonable, and unique disorders of physical objects has emerged [1], [2]. Since the physical unclonable function (PUF) is the best known security primitive in this category, we use the term PUF to generally refer to physical unclonable disorder-based security. The PUF satisfies many of the aforementioned requirements for physically securing modern and pending security applications which are not provided by classic cryptography. Furthermore, PUFs are based on elegant and intriguing concepts that are fascinating from both scientific and engineering points of view. The most popular PUFs exploit the escalating indelible process variation in modern silicon implementations of integrated circuits. Therefore, there has been a remarkable and exponentially growing interest in studying and fabricating PUFs.

Nevertheless, PUFs have several significant limitations. These limitations include the limitations of secret-key based protocols which were the targets of the original PUF technology, as well as unreliability in the presence of operational or environmental variations. Additionally, it has been demonstrated that several PUF structures are easily susceptible to complete reverse engineering or at least accurate prediction of the outputs once when enough challenge-response pairs become available or side channels are exposed [3], [4], [5], [6]. Our goal is to give an impetus to future research and development directions that would enable that the full potential of the PUF as a security primitive and that would enable the full realization of other related protocols such as those providing a public-key like functionality, a.k.a., physical unclonable functions (PPUF) [7], SIMPL [8], or the timed PUF [9].

Our paper provides a brief survey of several recent PUF-related developments and, to a serious extent, emphasizes a very subjective opinion about the most promising research and development directions for PUFs. To be specific, we believe that it is very important that the next generation of PUFs is digital, that it targets public key protocols, and is implemented in pending technologies. Technological developments are particularly important for PUF as the variance in technologies like FinFET and depleted silicon is expected to be reduced.

II. Security Paradigms for PUF Design

Paradigms can be defined as generic strategies that can be easily customized to address many problems in some scientific or engineering field. For example, dynamic programming and divide-and-conquer are effective optimization paradigms. Paradigms have been essential for the development of many research domains. Shannon created two security paradigms, diffusion and confusion, in his seminal paper in 1949 [10]. The confusion paradigm states that one should use highly non-linear computational elements. Obviously, the relevant observation is that it is easy to recover the inputs from the outputs if the computational elements are linear. The diffusion paradigm insists that each output should depend on as many inputs as possible at each computational stage. Surprisingly, both confusion and diffusion are rarely explicitly used in the creation of PUF structures. We expect that they will find more use in future PUF structures and analysis.

Recently, a new concept in PUF design has emerged that exploits nanotechnologies as the main enabler [11], [12], [13]. These nanotechnologies exhibit extremely non-linear output responses, thus satisfying Shannon's confusion principle.

978-3-9815370-2-4/DATE14/©2014 EDAA.

Furthermore, their bidirectionality enables conceptually new approaches to security protocols.

It is also important to consider PUF paradigms in public key security scenarios. There must be a special focus in not just creating protocols, but also ensuring that they are practical, i.e., that their energy and latency are low and their throughput is high while the required hardware is small. Public key PUF structures such as the PPUF, SIMPL, or the timed PUF also require their own paradigms. Initially, all of these methods were based on exploiting the gap between the execution time of the PUF circuitry and the required simulation time when the parameters of the pertinent PUF serve as the public key. However, in corresponding protocols, either one or both participating parties are required to conduct significant and energy intensive computational efforts, i.e. they have to either compute many challenge-response pairs or to conduct at least one long simulation. If the PUF public key has a suitable structure then the second requirement can be eliminated [14].

The requirement for significant computation has been eliminated in second generation PPUFs. This is achieved through the use of device aging and software disabling to reduce the computational requirement to only two clock cycles. The paradigm is that the delay of each transistor can be altered using device aging in such a way to create two identical PUFs after a subset of transistors is eliminated from the PUF using software techniques, such as special input selection [15], [16]. Much more research for implementing and evaluating this concept in current and pending technologies is still required.

Finally, digital PUFs require a completely new set of design and operational paradigms. One of them, bimodal functions, was recently proposed [14]. We expect that several conceptually new paradigms and specific techniques for digital public key PUFs will soon emerge.

III. Public Key Cryptography and New Security Applications

PUF is an excellent security primitive. For example, it operates at a low latency and high throughput and requires ultra low energy and power. Nevertheless, PUF has several serious limitations. Among the most restrictive is that the PUF targets secret key cryptography. Thus, its application domain is often restricted to authentication and random key generation.

The first important observation is that PUFs can enable effective techniques for other security tasks such as passive and active hardware intellectual property protection, software intellectual property protection, hardware obfuscation, prevention of reverse engineering, hardware intellectual property watermarking, remote trusted sensing and computation, software metering and the prevention of hardware Trojan embedding [2], [17]. PUFs can also enable important conceptually new security tasks such self trust which enables a user to trust herself. Another important application is that PUFs can be designed in such a way that they act as synchronized hardware random number generators [14], [16] or true-random number generators [18].

Public key cryptography is the crown jewel of classical cryptography. Recently, several techniques have been proposed that embed PUFs in security protocols for public key communication [7], [8], [9]. These implementations are intrinsically more resilient to attacks such as power and delay analysis attacks. They also enable a very low energy realization of public key protocols and can enable ultra fast protocols for many security protocols which are currently too slow to be practical [14], [15], [16]. There is also an ongoing effort to use PUFs to implement more complex security algorithms such as bit-commitment and oblivious transfer [19], [20]. We expect a great flurry of research activities that would further improve the initial efforts along these lines.

IV. Digital PUF

A primary initial motivation for the realization of the digital PUF is the natural requisite for both stability and an ultra low error rate in a hardware security primitive. The trend of moving circuits from the analog domain to the digital domain is well acknowledged and precedented across many disciplines, including radios, receivers, transmitters, and electronic components of discs.

The first generation of PUFs leveraged inherent process variation found in silicon implementation technologies for their creation. Therefore, it is not obvious at first that creating digital PUFs is even possible. Our main conjecture is that there are actually many mechanisms that still have to be discovered that will enable the realization of digital PUFs. Here, we briefly focus on two such mechanisms.

Recently, Xu et al. published the first digital PUF that uses the combination of a programmable FPGA-like fabric along with a standard analog PUF [14]. The idea is to use the standard analog PUF for the initialization of values in SRAM cells that dictate the functionality of the digital PUF. Before its first use, the analog part is intentionally maximally and rapidly aged in such a way that it becomes resilient to operational and environmental conditions. This step can be accomplished, for example, by using accelerated HCI-based aging. Once the digital SRAM-based PUF is activated, the accompanying analog PUF is effectively discarded. What is remarkable about this PUF is that it has a very small silicon footprint, it is ultra fast (only a few clock cycles), and it has very low energy consumption. However, as it is clear from its description, it is still not yet completely digital because it requires the initial use of an analog PUF at least for its first initialization.

More recently, Xu has developed a fully digital PUF. This PUF leverages the observation that a very small number of circuit faults can significantly impact the output of many pieces of digital logic in such a way that the correlation between the correct functioning and incorrect functioning of a faulty IC are very low. There are many ways in which these faults can be utilized and/or induced. For example, a significant percentage of circuitry in any aggressive technology has any number of significant faults. Another idea that further improves upon this observation and avoids potential problems with faults that are correlated on different ICs is to create gates and wires with a relatively high susceptibility for faulting. In these circuits, the creation of faults is a consequence of process variation. It has also been shown that these circuits also have low energy and power requirements, they are fast, and that they can pass the full set of NIST statistical randomness tests. Thus, this type of PUF can be used as a digital hardware random number

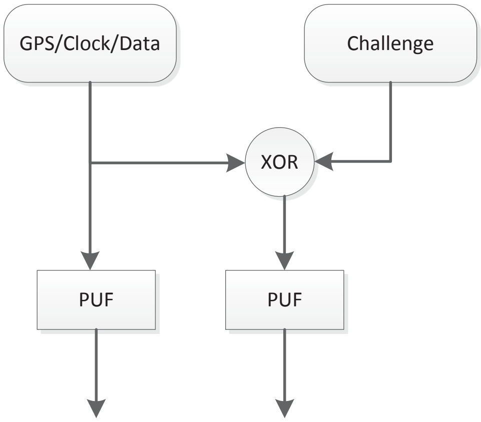

Fig. 1: Trusted remote computation flow [21]. A user-provided challenge is XORed with GPS, clock, and data and passed through the PUF as shown above. The responses are transmitted back to the remote user.

generator as well as support both symmetric and asymmetric cryptography.

One of the advantages of digital PUFs are along the lines of a low latency, high throughput, and low energy permutation of classical secret and public key cryptography protocols. We believe that it is of even higher importance that this circuitry can be used for the creation of secure trusted and private information flows in both classic and new security, trust, and privacy applications. In particular, they can be beneficial for tasks such as secure remote sensing.

Trust is an essential element for many applications, including social networking, cloud computing, and secure remote sensing. Trustable remote sensing is an activity in which a remote party can obtain proof that a particular sensor's recordings are indeed collected by that actual sensor, at a specific location and time. An example of a trusted remote computation flow utilizing PUFs is depicted in Figure 1.

V. Technology Impact on Next Generation PUFs

Numerous new technologies have been proposed for the implementation of various components of integrated circuits including memristors, resistance-based memory, and wireless on chip components. Each new technology may have a great impact on the implemented PUF structures and effectiveness.

The absence of process variation in a technology eliminates the possibility that popular and effective hardware-based primitives, such as PUFs and PPUFs, can be created. Note that randomness alone is not sufficient for the design of these types of hardware security primitives. The first problem is that process variation can follow a distribution with heavy tails. If this is the case, then a few gates will have either very long or very short delays. Thus, the attacker can easily guess the corresponding output, e.g, by observing the inputs that control only a few multiplexers. Let us assume the case of an unpredictable delay-based arbiter PUF, where the differences in delays for each stage is identical and there exists an equal number of faster and slower top and bottom branches [1]. In this case, a simple analysis can show that the variance and correlations in the underlying delay values weakens the overall security [22].

An important observation is that any analysis of the delay distribution for a particular technology is insufficient for creating high quality unpredictable PUFs. It is crucial that correlations are considered. For example, if there are strong correlations among variations on the same chip, then different chips might also be strongly correlated and thus, easily targeted for security attacks.

Another important observation is that one cannot consider a single technology's security properties in isolation. We illustrate this point by considering randomness due to process variation and due to device aging (introduced post-silicon.) If the impact of device aging is much higher than the impact of process variation, one can easily tune PUF delays such that large families of PUFs are unsusceptible to machine learning attacks. Furthermore, device aging can be used to render side channel attacks difficult by enabling the creation of two or more identical gates [23] even in the presence of process variation. If the situation is reversed and the impact of process variation is stronger than the impact of device aging, then the creation of matched PPUFs for ultra fast and ultra low energy security can be enabled. However, if the effects of process variation can be mitigated by device aging, then the complete notion of the matched PPUF would be eliminated. Other scenarios may also be created by restricting the aging patterns.

VI. PUF Testing and Metrics

One of the most effective ways to identify and develop effective cryptographical primitives and protocols is through extensive and creative analysis of existing proposals by other members of the community. The effectiveness of this approach has already been well demonstrated in PUF research. For instance, it has been demonstrated that several classes of PUFs can be broken surprisingly easily [3], [4], [5], [6]. For example, some of the earliest classes of PUFs, such as linear and feed-forward classes, have easily been broken using iterative linear programming or linear algebra and numerical analysis techniques. Furthermore, the prediction of PUF outputs has also been shown using machine learning techniques.

There are two conceptual problems with this approach. The first is that if a PUF passes one test it may fail another which better exploits a specific security weaknesses of the PUF. Furthermore, even if all tests pass, protection is not guaranteed due to the existence of side channel attack techniques which do not observe inputs or outputs. The second problem is that this approach does not establish high confidence that the PUF under test is indeed a high quality security primitive.

A very popular generic mathematical technique is to map an arbitrary instance of an existing problem into an instance of a new problem of interest. If the new problem can be solved well, then this indicates that the initial (existing) problem can also be solved well. For example, this procedure is commonly used in theoretical computer science to establish that a new problem of interest is NP-complete. In general, mapping is a powerful technique and often a practical way to address many new problems.

We propose to use numerous tests for random number generators for establishing the quality of a specific PUF [24]. For instance, if a PUF passes all NIST or DIEHARD tests for random number generators and the PUF is also broken, it would imply that the evaluating tests are also broken. That would be highly unlikely, but a powerful and broadly important result. Furthermore, since these tests are numeric, this technique allows that several competing PUFS can be compared.

There are two important related and relevant observations. The first is that the prediction of PUF outputs using inputs and other outputs is only one type of attack. While this attack is universal, it is easy to envision many other security threats. The second observation is that tests for random number generators may be overly demanding in the sense that they are indeed sufficient, but most likely not always necessary depending on the PUF's application to specific security protocols.

The MIT PUF group was the first to consider the use of random number generators for testing PUFs [1], in addition to the conventional intra-PUF and inter-PUF Hamming-distance based methods. They were able to show that their PUF is capable of passing the NIST statistical suite but only after complex feedback networks and the von Neumann trick were employed. The von Neumann trick forms from two signals, one that has a guaranteed perfect frequency (1/2) by eliminating all pairs of identical two signals and mapping signal pair (0,1) into 0 and (1,0) to 1 [25]. The NIST tests have also been applied to digital PUFs.

The following example shows that rather small changes in a PUF structure can impact the performance on the NIST tests. Table I depicts the results of the NIST statistical suite on two XOR network delay-based PUFs similar to the one depicted in Figure 2. PUF B corresponds to a PUF that has a lower branching factor between subsequent columns of XOR gates than PUF A. Note that the spectral test, measuring the periodic features in the bit stream, does not pass for all output streams generated by this PUF.

Figure 3 depicts the distribution of output hamming distances for pairs of inputs which differ by one hamming distance. While the standard delay-based PUF possesses many security properties, it does not satisfy the avalanche criterion that for a single bit flip of its input, the outputs should change dramatically [3], [4], [26]. The ideal case is that output hamming distances should follow a binomial distribution centered on half of the output vector length.

We conclude this Section by stating how the PUF's NIST tests should be organized. The outputs of the PUF under test should be sent back to the PUF as input in subsequent clock cycles (iterations). Ideally, this feedback should not contain any intermediate application of additional randomness. In any case, a source of overall randomness should not be hidden inside of the feedback network.

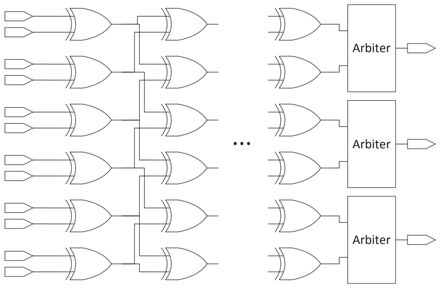

Fig. 2: Simplified XOR network delay-based PUF design.

TABLE I: NIST results for XOR network delay-based PUFs that differ in branching factors.

Statistical Test	Average Success Ratio	
	PUF A	PUF B
Frequency	97%	98%
Block Frequency (m=128)	100%	100%
Cusum-Forward	96%	98%
Cusum-Reverse	97%	97%
Runs	97%	99%
Longest Runs of Ones	94%	99%
Rank	98%	100%
Spectral DFT	95%	42%
Non-overlapping Templates ($m = 9$)	97%	98%
Overlapping Templates ($m = 9$)	99%	98%
Universal	100%	100%
Approximate Entropy ($m = 10$)	75%	75%
Serial ($m = 16$)	99%	99%
Linear Complexity ($M = 500$)	95%	97%

VII. The Quest for Efficient PUF Protocols

The unpredictability of PUF responses to random challenges gives us the ability to efficiently implement security protocols, like authentication, with PUFs. These protocols are usually executed between a party with access to a physical PUF and a trusted party who has access to the PUF compact model. Every such protocol based on the PUF should be able to take in to account the inherent instability of PUF responses due to the environmental noise such as temperature variations.

In the context of challenge-response pair authentication, sending error correction syndrome bits for correcting the errors before a hash operation was proposed in [1]. A newer Index-Based Syndrome (IBS) error correction coding for PUFs has also been introduced in [27].

The aforementioned protocols based on error correction incur a high power and area overhead, which impedes their deployment in ultra low power applications. An alternative scheme was proposed in [28] that uses pattern matching of PUF responses to generate secret keys. Authors in [29] expanded the idea of pattern matching of PUF responses to implement an ultra lightweight and secure authentication protocol. This protocol, dubbed Slender PUF, is based on covert indices that can be recovered by pattern matching. In this protocol, the owner of the PUF, randomly selects and sends a subset of PUF challenge-responses, so the exact position of the subset in the PUF challenge-response stream is obfuscated from eavesdroppers. Authentication of the responses is performed by matching the substring to the available full response string.

Recently proposed memristors and III-V nanowires are susceptible to process variations akin to traditional CMOS process. Therefore, they can be utilized to implement interesting types of PUFs. Besides process variations, memristors and nanowires also have a bidirectional and non-linear input-

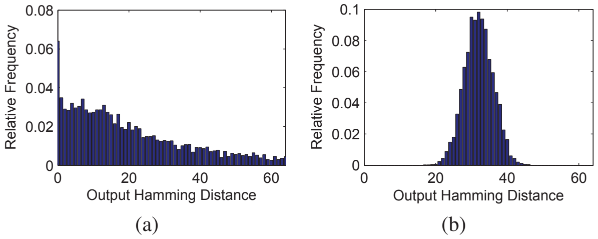

Fig. 3: The avalanche effect of (a) standard delay-based PUF and (b) digital PUF [14]. Output hamming distance is measured between two outputs when their corresponding input vectors differ by only one hamming distance (i.e. one bit).

output behavior, which has enabled researchers to propose innovative PUF circuitry based on them [11], [30]. A very efficient authentication and key-exchange protocol based on these novel structures has been proposed in [12].

VIII. Attacks and Countermeasures

A. *Rising Attacks*

The most nefarious attack against physical-based disorder security is cloning. For both weak and strong PUFs, cloning can be done by reverse-engineering (RE) [2]. The security of these PUFs centers on the concealment of the underlying random (unclonable) circuit variations. For weak PUFs, the adversary can learn the bounded number of secret values by RE and then clone the PUF by storing the learned values in another memory. For strong PUFs, RE of the internal structure of the pertinent circuitry provides a basis for cloning in a compact way, without the need for storing a huge database of CRPs.

For unique objects (UNOs), where security is based on extraction of static analog fingerprints of the medium, RE can be also used for cloning purposes. As several such analog UNO structures have recently been identified [2], RE attacks on those structures is a normal topic that would follow.

There are two distinct directions for pursuing research in RE: (i) noninvasive methods, and (ii) invasive techniques. The former category can be realized in various ways, e.g., by passive/active eavesdropping of the CRPs to/from the PUF and then using a data analysis/machine learning method to model the physical PUF such as [3], [4], [5]; another noninvasive attack performs measurements on the device in operation, such as image or power *side-channels* and then integrates the measured values into the data analysis algorithm [6]. Both noninvasive methods exploit the correlations between the available information (from the side-channels or from eavesdropping) and the secret structure/values associated with PUFs.

Much more exciting research is under way in the noninvasive RE direction, including cloning the newer PUF structures, obtaining novel side-channel information, and innovations in data analysis. For example, although current machine learning methods have proven to be effective for RE of certain PUF structures, it is not immediately evident if they are directly applicable to newer structures. Furthermore, data analysis and machine learning require the art of selecting the relevant method: even if a method of choice does not help with modeling the PUF, it is mostly unclear if other methods cannot model the PUF. Therefore, investigating applicability of efficiency of innovative data analysis and machine learning methods is a timely topic.

In terms of the latter category of invasive attacks, which is often destructive, having access to the (rather) costly machinery for low-level debugging of ICs is required. As those machines are becoming more available with a higher resolution and at a lower cost, so is the susceptibility to this class of attacks.

RE, in a classic PUF sense, is not directly applicable to cloning PPUFs as the internal circuit structure of the device is assumed to be publicly available. For PPUFs, security is based upon the speed of computation of the original device; this speed should not be mimicked in simulations or emulations [31], [32]. Therefore, a wicked adversary must focus on finding software or hardware structures that can find the computation results within the bounded computation time of the original device.

Perhaps because of the sparse number of physical implementations of PPUFs and the novice nature of the topic compared to (weak/strong) PUFs, a comprehensive treatment of the cloning attacks against the PPUF is yet to be done. Of particular interest would be investigating the applicability of finite state and control flow models to each class of PPUF and if applicable, defining the underlying serial and parallel computations. Naturally, parallel computations can admit a concurrent implementation; the serial computations would determine the limit of timing in terms of the number of cycles. Note that, it is our belief that the research in PPUFs is still in its infancy; alternative ways for emulating of simulating PPUF CRPs may be also explored.

B. *Pending countermeasures*

Alongside the evolvement of attacks, corresponding countermeasures will also be devised. This in turn enriches and strengthens the attack possibilities. A set of effective countermeasures for both invasive and noninvasive attacks is based upon physical access restriction. For example, for weak SRAM PUFs, the security is solely based on the inaccessibility of the bits at the start-up state and inability to alter this state. Relevant countermeasures against physical and invasive attacks that need to be further investigated are tamper-resistance and tamper-proof technologies. Tamper resistance hardens depackaging and disintegration of the product, while tamper proof damages the device/content upon sensing of a tamper, so the device is rendered useless and noninformative for RE purposes.

Another important class of countermeasures that needs to be enriched are those that deter noninvasive attacks based on machine learning. This shall be done by breaking all possible patterns and correlations between the CRP set and the structural characteristics of the PUF, both in terms of the inter- and intra- CRP relationships [26], [4]. While the first order correlations are usually the only metrics studied for patterns, the higher order correlations need to be detected and trimmed.

There are two classes of countermeasures that are typically used for deterring the side-channel attacks by breaking the correlation between the side-channel structure and the circuit operation. The first class of methods is based upon reduction or elimination of the leaked information. The second class of methods is centered on lessening or removing the relationships between the secret data and the emissions from the leaked side-channel. Such lessening or removal is usually performed by randomization or by keeping the side-channel value constant [33]. For thwarting each possible side-channel, one or more appropriate measures has to be devised accordingly.

Note that our list of pending countermeasures in this section is meant to be representative, and not exhausting. It is possible to create countermeasures at different levels of the stack, other than the device stack, in particular for attacks that are not based on physical properties, e.g., eavesdropping. For instance, [29] has achieved resiliency against machine learning attacks by using substring-matching.

IX. Conclusion

PUF is an exciting, effective, and elegant hardware security primitive that has attracted a great deal of research attention. We provide a brief overview of PUF research with complete emphasis on very recent developments and most challenging and potentially most rewarding future research directions. Specifically, we cover tasks, challenges, and opportunities related to digital PUFs, sound security paradigms, new public key tasks, quantitative metrics for PUF evaluation, and the impact of new technologies, rising attacks, and pending countermeasures.

X. Acknowledgments

This research is in part supported by Army Research Office (ARO) YIP award under grant No. R17450 and a NSF grant to Rice University (CNS-1059416). It is also supported by NSF grants (CNS-0958369, CNS-1059435, and CCF-0926127) and Air Force Award to UCLA (FA8750-12-2-0014).

References

[1] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in *ACM Conf. on Computer and Communications Security*, pp. 148–160, 2002.

[2] U. Rührmair, S. Devadas, and F. Koushanfar, "Security based on physical unclonability and disorder," *Book Chapter in Introduction to Hardware Security and Trust*, 2011.

[3] M. Majzoobi, F. Koushanfar, and M. Potkonjak, "Testing techniques for hardware security," in *Int. Test Conference (ITC)*, pp. 1–10, 2008.

[4] M. Majzoobi, F. Koushanfar, and M. Potkonjak, "Techniques for design and implementation of secure reconfigurable PUFs," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 2, no. 1, 2009.

[5] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber, "Modeling attacks on physical unclonable functions," in *ACM Conf. on Computer and Communications Security*, pp. 237–249, 2010.

[6] A. Mahmoud, U. Rührmair, M. Majzoobi, and F. Koushanfar, "Combined modeling and side channel attacks on strong PUFs," *IACR Cryptology ePrint Archive*, no. 632, 2013.

[7] N. Beckmann and M. Potkonjak, "Hardware-based public-key cryptography with public physically unclonable functions," in *Information Hiding*, pp. 206–220, Springer, 2009.

[8] U. Rührmair, "SIMPL systems: On a public key variant of physical unclonable functions," *IACR Cryptology ePrint Archive*, no. 255, 2009.

[9] M. Majzoobi and F. Koushanfar, "Time-bounded authentication of FPGAs," *IEEE Trans. on Information Forensics and Security (TIFS)*, vol. 3, no. 6, pp. 1123–1135, 2011.

[10] C. E. Shannon, "Communication theory of secrecy systems," *Bell system technical journal*, vol. 28, no. 4, pp. 656–715, 1949.

[11] J. B. Wendt and M. Potkonjak, "Nanotechnology-based trusted remote sensing," in *IEEE Sensors*, pp. 1213–1216, 2011.

[12] J. B. Wendt and M. Potkonjak, "The bidirectional polyomino partitioned PPUF as a hardware security primitive," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2013.

[13] G. S. Rose, J. Rajendran, N. R. McDonald, R. Karri, M. Potkonjak, and B. T. Wysocki, "Hardware security strategies exploiting nanoelectronic circuits," in *ASP-DAC*, pp. 368–372, 2013.

[14] T. Xu, J. B. Wendt, and M. Potkonjak, "Digital bimodal function: an ultra-low energy security primitive," in *Proc. of Int. Symp. on Low Power Electronics and Design*, pp. 292–296, 2013.

[15] S. Meguerdichian and M. Potkonjak, "Matched public PUF: ultra low energy security platform," in *Proc. of Int. Symp. on Low Power Electronics and Design*, pp. 45–50, 2011.

[16] S. Meguerdichian and M. Potkonjak, "Using standardized quantization for multi-party ppuf matching: Foundations and applications," in *Int. Conference on Computer-Aided Design*, pp. 577–584, 2012.

[17] F. Dabiri and M. Potkonjak, "Hardware aging-based software metering," in *Proc. of the Conf. on Design, Automation and Test in Europe*, pp. 460–465, 2009.

[18] M. Majzoobi, F. Koushanfar, and S. Devadas, "FPGA-based true random number generation using circuit metastability with adaptive feedback control," *Cryptographic Hardware and Embedded Systems*, pp. 17–32, 2011.

[19] U. Rührmair, "Oblivious transfer based on physical unclonable functions," in *Trust and Trustworthy Computing*, pp. 430–440, Springer, 2010.

[20] U. Rührmair and M. van Dijk, "On the practical use of physical unclonable functions in oblivious transfer and bit commitment protocols," *Journal of Cryptographic Engineering*, pp. 1–12, 2013.

[21] M. Potkonjak, S. Meguerdichian, and J. Wong, "Trusted sensors and remote sensing," in *IEEE Sensors*, pp. 1–4, 2010.

[22] W. Feller, *An introduction to probability theory and its applications*, vol. 2. John Wiley & Sons, 2008.

[23] K. Tiri and I. Verbauwhede, "A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation," in *Proc. of the Conf. on Design, Automation and Test in Europe*, pp. 246–251, 2004.

[24] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, and E. Barker, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," tech. rep., DTIC Document, 2001.

[25] J. Von Neumann, "Various techniques used in connection with random digits," *Applied Math Series*, vol. 12, no. 36-38, p. 1, 1951.

[26] M. Majzoobi, F. Koushanfar, and M. Potkonjak, "Lightweight secure PUF," in *Int. Conference on Computer-Aided Design (ICCAD)*, pp. 670–673, 2008.

[27] M.-D. M. Yu and S. Devadas, "Secure and robust error correction for physical unclonable functions," *IEEE Design and Test of Computers*, vol. 27, pp. 48–65, 2010.

[28] Z. Paral and S. Devadas, "Reliable and efficient PUF-based key generation using pattern matching," in *Int. Symp. on Hardware-Oriented Security and Trust*, pp. 128–133, 2011.

[29] M. Majzoobi, M. Rostami, F. Koushanfar, D. S. Wallach, and S. Devadas, "Slender PUF protocol: A lightweight, robust, and secure authentication by substring matching," in *IEEE Symp. on Security and Privacy Workshops*, pp. 33–44, 2012.

[30] J. Rajendran, G. S. Rose, R. Karri, and M. Potkonjak, "Nano-PPUF: A memristor-based security primitive," in *Computer Society Annual Symp. on VLSI*, pp. 84–87, 2012.

[31] U. Rührmair, "SIMPL systems, or: can we design cryptographic hardware without secret key information?," in *SOFSEM 2011: Theory and Practice of Computer Science*, pp. 26–45, Springer, 2011.

[32] M. Potkonjak, S. Meguerdichian, A. Nahapetian, and S. Wei, "Differential public physically unclonable functions: architecture and applications," in *Design Automation Conference (DAC)*, pp. 242–247, 2011.

[33] M. Rostami, F. Koushanfar, J. Rajendran, and R. Karri, "Hardware security: Threat models and metrics," in *Int. Conf. on Computer-Aided Design*, 2013.