Physically Unclonable Functions:
A newcomers perspective

Jaimes Joschko
Electrical and Computer Engineering
University of Victoria
Victoria, British Columbia

***Abstract*—In this report I give a general overview of my understanding of PUF technology, reviewed some of the literature, explored some the recent work in PUFs, and give some potential improvement ideas I think are worth exploring.**

***Index Terms*—Physically Unclonable Functions (PUF), Integrated Circuits, Physical Security, Embedded Security, Computer Security.**

I. Introduction

Software can no longer naively rely on hardware as a root of trust. With attacks such as side-channel analysis, hardware trojan insertion, and reverse engineering it is clear that any software security control can be circumvented if the underlying hardware is compromised. Lots of work has been done to secure hardware and re-establish it as a root of trust. One promising technology that can help implement controls and secure hardware is the Physically Unclonable Functions (PUF).

II. Background

According to [1], [2], [3], [4] research on what has become known as PUFs began in the early 2000s with [5], [6], [7], [8]. Since then, there has been many papers and some good books on PUFs, their designs, and their applications. A good timeline of the developments of PUFs can be found in [1]. In the following subsections I will give a general overview of Physically Unclonable Functions.

A. General Definition of a PUF

Physically Unclonable Functions (PUF) are hardware security primitives which can generate device specific fingerprints that are derived from the unique physical variations and disorder inherent in all physical objects [2]. An Alternate definition is that a PUF is a physical entity whose behaviour is a function of its design structure and the physical disorder derived from its manufacturing process [3]. Abstractly, a PUF instance i can be thought of as blackbox that maps a set of challenges and device specific physical variations to a set of responses. This mapping determines the Challenge-Response Pairs (CRPs) of the PUF instance i. The set of all CRPs for a given PUF instance i is called the challenge-response behaviour of i [4].

Ideally, a PUF is an easily evaluable cryptographically secure one-way function whose device specific challenge-response behaviour depends on the the unique physical variations and disorder inherent in the physical device [1], [2]. This means that the challenge-response behaviour of an ideal PUF is unpredictable but deterministic, and for each PUF instance the challenge-response behaviour depends on unique physical variations inherent in the device it is embedded in. Because it's challenge-response behaviour depends on the unique physical variations inherent in the device, each ideal PUF instance is physically unclonable. Therefore, ideal PUFs are well suited to provide device specific fingerprints and act as hardware security primitives. Of course the real world is not ideal, so trade-offs and considerations are required in order to develop suitable PUFs.

B. Types of PUFs

PUFs can be constructed for nearly all physical objects by using the inherent properties of the object. This is because physical disorder is present in all object, even in highly controlled products like integrated circuits (ICs). The main types of PUFs are: Optical PUFs, Acoustic PUFs, Radio-Frequency PUFs, Coating PUFs, Surface-Based PUFs, and Silicon PUFs [1], [4].

Optical PUFs rely on the random scattering of light when an object's particles are radiated with a laser. The scattered light can be measure by analyzing the speckle pattern, essentially an image of the scattered light, which is highly random and unique to each object [1].

Acoustic PUFs are similar to Optical PUFs but use sound waves that propagate through a solid medium and randomly scatter on the particles in the medium. This scattering of the sound waves can be captured using a transducers to measure vibrations [1].

Similarly, Radio-Frequency PUFs use electro-magnetic wave emanations to characterize an object [4]. This can be thought of a beneficial use of side-channels in the case of ICs.

Coating PUFs use embedded sensors to measure the electrical properties of a special protective coating applied to the outside of an object [1]. Due to the random distribution of doping particles the response measured by the sensors will be unique. Furthermore, attempts to physically tamper with the object will change the coating's electrical properties. Thus, coating PUFs provide a level of protection again physical attacks [1].

Similar to Coating PUFs, surface PUFs exploit the microscopic imperfections on an object's surface to extract unique

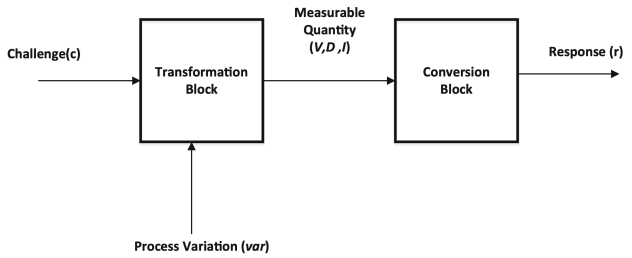

Fig. 1. the generic structure of a silicon PUF.[4]

identifying information[4]. For example, two seemingly identical pieces of paper can be distinguished using the microscopic imperfections in their surfaces. In this case the "challenge" is looking at the surface with a microscope and the "response" is the pattern on the surface. Multiple CRPs can be created by sampling different regions of the surface.

Finally, Silicon PUFs use the uncontrollable physical disorder inherent when manufacturing ICs [1], [4]. This disorder causes identically designed circuits with identical masks to have different electrical properties. Thus, by triggering this type of circuit and measuring the output, a unique challenge response behaviour can be extracted for each circuit. Therefore, instantiating an easily evaluable instance of a PUF in each IC.

Silicon PUFs will be the focus of the rest of the report. Although, I believe it is an interesting question if multiple types of PUFs can be combined to solve an outstanding problem or enhance the effectiveness of a given PUF application?

C. Silicon PUFs

As IC circuits scale down, quantum effects become more pronounced [4]. Furthermore, during manufacturing, exact device/interconnects geometry and device/interconnects material uniformity becomes exponentially harder to control [4]. These variations all effect the exact electrical properties (i.e. current, delay, and voltage) of the IC and can be harnessed to create PUFs.

A very useful diagram to keep in mind when thinking about silicon PUFs is Figure 1 taken from [4].

There are many ways to characterize silicon PUFs. One is by their realizations in hardware. The two main categories in this flavour of characterization are *Delay Based PUFs* and *Memory Based PUFs*, with many different architectures of each type [1], [3], [4].

1) Delay Based PUFs: Arbiter PUFs in general consist of a sequence of switches in series and an arbiter at the end of the sequence. The challenge to the arbiter PUF uses the switches to select two symmetrical, parallel and identically designed delay lines. Because of process variations, the delays in the two selected lines will be different even though they are identically designed. A signal is raced along the lines and the arbiter selects the winner. The response is either a zero or a one depending on which line wins the race. If there are n switches, then the number of CRPs is 2^n. Each CRP is a n-bit string that selects the paths and a 1-bit response [1], [2], [3], [4]. Note: Longer response words can be constructed by sampling the PUF multiple times with different challenges. Another interesting technique is using the response as a seed for a pseudo-random number generator.

Lightweight PUFs often use the same basic design idea as Arbiter PUFs, but try to introduce non-linearities and other complications to provide resilience against modelling attacks. To introduce non-linearities and other complications, paths are fed forward or split, the challenge is broken up and applied to multiple PUFs and finally the responses are combined in a way that further obfuscates delay relationships. It is unclear if these designs succeed in protecting against current machine learning modelling attacks [2], [3].

A ring oscillator (RO) is a circuit made up of an odd number of inverters in a loop. Since there are an odd number of inverters, when a signal is applied to the loop, it will oscillate at a frequency which is in part dependant on the delay variations of the inverters and lines in the loop. Ring Oscillator PUFs implement n identically designed ROs and select two to compare. Because of variations in the manufacturing process each ROs will have a slightly different oscillating frequency. The challenge is a $\log(n)$-bit string that selects two ROs, and the response is a zero or one depending on which RO has the higher frequency. The number of challenge response pairs is $\frac{n\times(n-1)}{2}$ (i.e. the number of possible pairs among n ROs) [1], [2], [3], [4]. RO PUFs are very popular do to the ease in which they can be incorporated into hardware designs.

Other types of common delay based PUFs are the Bistable Ring PUF, Glitch PUF, ALU PUF and Loop PUF.

2) Memory Based PUFs: SRAM PUFs exploit the random but often stable behaviour of static random-access memory (SRAM) cell upon initial startup. Some cells will self initialize to 1 and others to 0 when initially started up due to random physical variations. These start up values are stable across multiple startups, but are unique for each device. Thus, a PUF can be created by sending a challenge to select a region of SRAM cells after startup but before the operating system (OS) initializes the memory. The response is the value of the sampled memory addresses [1], [2], [3], [4].

Other common types of memory PUFs, that are similar to the SRAM PUF are Butterfly PUF, Flip-flop PUF, and Latch PUF.

D. Desired Properties

Ideally, a silicon PUF should have a simple and small architecture in order to be useful in resource constrained embedded systems. It should be easy to manufacture and not require custom design software to integrate it into the overall IC design in order to limit costs. To be practical, the PUF needs to be easy to evaluate and reliable in diverse operating conditions. Finally, the PUF needs to be secure against physical, side-channel, and modelling attacks in order to be useful. To insure these properties hold, performance metrics are used.

E. Performance Metrics

To help measure and compare designs the following properties are often used [1], [4], [3].

- Simplicity, Size, Complexity
  These properties are often used since one of the main applications of PUFs are in resource constrained embedded systems.
- Robustness, Reliability, Reproducibility
  Since silicon PUFs rely on tiny variations, their CRPs are highly susceptible to changes in operating conditions and IC aging. To characterize a particular PUF architecture's ability to resist changes in operating conditions and IC aging these properties are used. Note: often PUFs preform poorly on these metrics and need to augmented by redundancy, error correcting codes, helper data, or pre-aging techniques.
- Unpredictability, Uniformity, One-Wayness
  To be useful in authentication, identification, key generation, and other applications PUFs are measured against these properties. If they do not sufficiently demonstrate these properties then the design is likely vulnerable to modelling attacks.
- Uniqueness, Physical Unclonability
  PUFs CRB should be unique for each device even if two devices use the same PUF design and mask, and it should be infeasible to make physical copies of a particular device's PUF even if the architecture is known.
- Tamper-evident, Tamper-resistance
  PUFs CRB should be highly sensitive to tampering and reverse engineering in order to be resistant to physical attacks and reliable in protecting intellectual property.

The above properties are explained in detail, along with ways to quantitatively evaluate them, in [1], [3], [4]. For example, Uniqueness can be measure using the inter-chip hamming distance and Reliability can be measure using the intra-chip hamming distance [1], [3], [4].

F. Applications

1) Authentication: PUF are good candidates for light weight authentication in platforms with limited resources such as RFIDs or in resource constrained embedded systems. The most basic scheme involves a PUF being included in the design and manufacturing of a device and then before deployment a trusted third party queries the PUF to construct a database of the CRPs for the PUF. The PUF can then be challenged in the field with one or more challenges from the CRP database and authenticated if it responds with enough correct responses [1], [2].

2) Secure Key Storage/Generation: Instead of storing secret keys used in cryptographic protocols in flash or other memory that can be read out by attackers, PUF instances can be used to "store" the key in "hardware" and generated it on the fly. A set of challenges are applied to the PUF and the key is constructed from the responses. This helps protect against physical attacks due to the Tamper-evident/Tamper-resistance properties of PUFs. Any physical intervention will alter the variations in the PUF and thus corrupt the key. A significant issue is insuring that the PUF reliably produces the same key over time and different operating conditions. There are many techniques but often these can result in reduced entropy or opens vulnerabilities to modelling attacks [1], [2], [3], [4].

3) Remote Secure Sensors: Security is often an import consideration for sensor networks. An interesting application of PUF technology is in securing remote sensors from eavesdropping or interference. Often sensors are small and too resource constrained to be deployed with encryption technology. Instead, remote secure sensor PUFs exploit the sensitivity of PUFs to environmental factors and "encrypt" signals through the change in their CPB. The deployer of the sensors creates a database of CRPs and also keeps track of how these CRPs are effected by environment factors (i.e. temperature, etc.). Then, when they want to sample the sensor, they send a challenge to the PUF and compare the response to the set of responses in the database. All an eavesdropper sees is a challenge response pair but can't infer the sensed physical quantity [4].

4) Other Applications: There are lots of other applications for PUFs. Some of which are Key Zeroization, Intellectual Property Protection, Counterfeit Protection, Hardware Metering, IC Identification, Remote Attestation, and Binding Software to Hardware [1], [2], [3], [4]. All of these application use some combination of the properties outlined above to achieve their goals. Unfortunately, the achievement of these properties is often imperfect. Thus, there have been many proposals for attacks against PUFs.

G. Attacks

The attacks on PUFs tend to break down into the following categories [1], [3].

1) Invasive attacks: Sometimes by increasing the reliability and robustness of a PUF the tamper-evident and tamper-resistance properties of a PUF design are reduced. Thus, physical attacks that clone a PUF or extract the secret information become possible [3].

2) Fault Injection Attacks: These attacks try to disrupt the normal CRB of a PUF by introducing adverse operating conditions [1]. These attacks can be used to conduct denial of service attacks or as part of a larger scheme to compromise security.

3) Side-channel Attacks: Side-channel attacks exploit timing delays, electro-magnetic radiation leakage, or other physical phenomena to extract secret information. This can be used to extract the secret key generated by a PUF or as a stepping stone in modelling attacks. As noted in [3], [4] lots of PUF designs are often vulnerable to side-channel attacks and more work needs to be done in this area.

4) Modeling Attacks: Modelling attacks are the most common attacks on PUFs. They consist of using techniques such as machine learning to constructing a mathematical model of the PUF, and then emulating the CRB of a PUF. For example, in a simple arbiter PUF delays add linearly and after observing enough CRPs an attacker can predict the responses to new

challenges with high probability. This in turn compromises the root of trust security desired by the PUF because an attacker can then clone the PUF in software and masquerade as the PUF. There has been lots of work on designing PUFs that are resistant to machine learning emulation attacks[1], [2], [3], [4].

III. Literature Review

In [5] they observe that as transistors get smaller, the effects of small variations (caused by the random distribution of dopant atoms) inherent in the transistors becomes more pronounced and that although this may be a problem it also presents an opportunity to uniquely identify Integrated Circuits. The process they describe, Integrated Circuit Identification (ICID), involves selecting a set of MOSFET (i.e. transistors) called ID cells then reading and converting the random voltage variations for the ID cells into a binary string. This binary string is then used to distinguish ICs with very high reliability.

In [6] they introduce the idea of a PUF, and give definitions of what a PUF is and explanations of how they work. They show that a PUF can be used to identify and authenticate ICs by extracting the manufacturing process variation in identically masked IC to construct unique sets of CRPs for each IC. These CRPs can then be used in standard, or more complex, authentication and identification schemes. The architectures they use to implement their idea of a PUF are pretty much the ring oscillator PUF and arbiter PUF described above. They also introduce modifications, such as hashes of challenges and responses, to make them more resistant to attacks and error correction to make the CRB more reliable.

In [7] they come at the idea of PUFs from a different direction. They consider PUFs as one-way functions that are based not on number theoretic assumptions, such as the discrete-log problem, but on the physical disorder inherent in objects. Unfortunately, I was only able to obtain a partial version of their report, but in it they explore the idea of using speckle patterns to identify and authenticate objects. They also describe the difficulty of emulating these patterns.

In [8] the authors are concerned about attacks enabled by techniques such as micro-probing and power analysis that are able to extract the secret keys from Smartcards and ATMs. They proposed the arbiter PUF to defend against these attacks. Furthermore, they implemented a test arbiter PUF and conducted experiments to measure what amounts to the intra-chip hamming distance being very good for this type of PUF. They also construct a model that helps measure the delay variations inherent in a given technology [8].

[1] and [2] provide very accessable and quick overviews of PUFs. On the other hand [3] and [4] go into much more depth about measuring PUF properties, compensating for aging and environmental noise, applications, architectures, and attacks on PUFs. For anyone looking to start understanding PUF technology [1], [2], [3], [4] are great resources.

IV. Related Work

In [9] they observe that RO PUFs are among the most reliable silicon PUFs because the frequency differences in the pair of ROs can be measured more accurately if the counters measuring their frequencies are left to increment longer. Thus, errors in the CRB due to oltage variations are reduced. Unfortunately, RO PUF are still sensitive to temperature variations. So, they propose a new "hybrid" RO PUF that incorporates current starved inverters which turn out to decrease the temperature sensitivity and power consumption of the PUF.

In [10] they propose the Differential Circuit PUF (DiffC-PUF) which is a hybrid of a silicon based PUF core and printed components which has advantages for reading out responses.

In [11] they present a hybrid delay based Arbiter Ring Oscillator PUF (AROPUF) which appears to be more resistant to machine learning attacks than the traditional arbiter PUF. Essentially they challenge a arbiter PUF and then use the arbiter PUFs response to select the ROs in a RO PUF. This helps to introduce more randomness and nonlinearities which makes machine learning attacks hard despite the CRB remaining relatively stable.

V. Ideas

A. Hybrid PUF Architectures

Reading through the literature I was a bit surprised that I couldn't find more Hybrid PUFs. As described in [3], [4] different PUF architectures perform better or worse on the above performance metrics. For example, delay based PUFs tend to be more unique and tamper resistant than memory based PUFs, but are more susceptible to voltage and temperature variations than memory PUFs. Also, delay based PUFs tend to be more susceptible to modelling attacks than memory based PUFs, but can have a much large challenge-response space for a given area than memory-based PUFs [3], [4]. So, why not try to get the best of both worlds by combining PUF architectures? The closest work I found to this idea was [11]. Clearly, the work in [9] and [11] could be combined to create a reliable and model resistant delay based PUF. But there are lots more possibilities to explore.

For example, consider the traditional arbiter PUF. It has a challenge-response space of 2^n. Ideally, we would like to have that challenge-response space for RO PUFs instead of $\frac{n\times(n-1)}{2}$ without drastically increasing the space used by the PUF. I am new to hardware design, but naively it seems like we could take the traditional arbiter PUF design and after every switch introduce two inverters. Then, have a loop back line that goes to the beginning of the series of switches. Counters could also be added. If there are an odd number of these switch-inverter cells then it seems like the PUF could operate as either an arbiter PUF depending on which line wins the race or as a RO PUF with 2^n CRPs but with only small increases in area used by the circuit. I don't have the background to explore this idea yet.

In a similar style as [11] it seems like a memory PUF's response could be used to challenge a delay-based PUF or vice versa. I think this would be an interesting design space to explore.

B. Zero-Knowledge Authentication Scheme

Another idea I had while researching and learning about PUFs is that zero-knowledge proofs could be incorporated into the authentication schemes. One big problem with traditional PUF authentication schemes is that CRPs cannot be reused since attackers could just do replay attacks. There are schemes that incorporate "helper data" which limits the information gained by attackers and can help increase the reliability of the CRB, but large databases of the CRPs are still required to be collected before deployment. Furthermore, to make schemes more resistant to modelling attacks hashes or encryption techniques are used which greatly increases the circuitry. If you can afford hashes in the circuitry than I think it may be possible to develop a zero-knowledge scheme which allows the PUF to prove it knows the response to a given challenge without revealing the actual response. This would allow CRPs to be reused.

VI. Conclusions

In this report I gave a general overview of my understanding of PUF technology, reviewed some of the literature, explored some the recent work in PUFs, and gave some potential improvement ideas I think are worth exploring.

References

[1] D. Mukhopadhyay and R. S. Chakraborty, *Hardware Security: Design, Threats, and Safeguards*. Boca Raton, FL: CRC Press Taylor & Francis Group, 2015.

[2] *Hardware Security: a Hands-on Learning Approach*. Cambridge, MA: MK Morgan Kaufmann Elsevier, 2019.

[3] C. Wachsmann and A.-R. Sadeghi, *Physically unclonable functions (PUFs): applications, models, and future directions*. San Rafael, California: Morgan & Claypool Publishers, 2015;2014;, vol. 12;12.;.

[4] B. Halak and S. O. service), *Physically Unclonable Functions: From Basic Design Principles to Advanced Hardware Security Applications*. Cham: Springer International Publishing, 2018.

[5] K. Lofstrom, W. R. Daasch, and D. Taylor, "Ic identification circuit using device mismatch." IEEE, 2000, pp. 372–373.

[6] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions." ACM, 2002, pp. 148–160.

[7] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions," *Science (American Association for the Advancement of Science)*, vol. 297, no. 5589, pp. 2030–2029, 2002.

[8] D. Lim, J. W. Lee, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas, "Extracting secret keys from integrated circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 13, no. 10, pp. 1200–1205, 2005.

[9] Y. Cao, L. Zhang, C. Chang, and S. Chen, "A low-power hybrid ro puf with improved thermal stability for lightweight applications," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 7, pp. 1143–1147, July 2015.

[10] L. Zimmermann, A. Scholz, A. Sikora, and J. Aghassi-Hagmann, "A hybrid system architecture for the readout of a printed physical unclonable function," in *2018 International Conference on Electronics Technology (ICET)*, May 2018, pp. 11–14.

[11] N. Pundir, N. A. Hazari, F. Amsaad, and M. Niamat, "A novel hybrid delay based physical unclonable function immune to machine learning attacks," in *2017 IEEE National Aerospace and Electronics Conference (NAECON)*, June 2017, pp. 84–87.