Physically Unclonable Functions found in Standard Components of Commercial Devices

André Schaller
Technische Universität Darmstadt (CASED)
Darmstadt, Germany
http://www.cased.de

Vincent van der Leest
Intrinsic-ID
Eindhoven, The Netherlands
http://www.intrinsic-id.com

I. Introduction

Physical(-ly) Unclonable Functions (PUFs) have become a very popular topic in both the scientific community and industry over the last few years. Many studies have shown the added value of PUFs for generating a hardware specific "fingerprint", which can be the basis for security implementations such as secure key storage and device authentication. Up to now most research has considered PUFs to be a security building block, which has to be added during the design phase of a chip (either ASIC or FPGA) before it can be utilized. However, many commercially available devices already contain standard components that might be suitable for PUF use. To investigate whether it is possible to find and utilize existing PUF instantiation in commercially available hardware, the European Commission has funded the PUFFIN project [1] as part of the FP7-programme. This project intends to study and show the existence of PUFs in standard PCs, laptops, mobile phones and other consumer electronics. The goals of the project are:

1) Uncovering PUFs in standard components, such as graphical processing units (GPUs), central processing units (CPUs) and SRAM start-up patterns, mainly focusing on the latter [2].
2) Evaluating the performance (robustness and randomness) of the discovered PUFs.
3) Studying and solving specific use cases, which show the added value of publicly available PUF instantiations.

In this "work-in-progress" paper, we will shine a light on some of the use cases that are currently being studied in the PUFFIN project. These use cases will demonstrate the results of the project with the goal for an open platform (being the most difficult element to secure in an information-technology system today) to inherit security properties from its own identity and its intrinsic physical properties.

II. Use cases for PUFs in commercial devices

In this section we will describe the most important use cases for employing intrinsic PUFs. We will demonstrate how we are looking at these (already known) use cases from a new angle, since we are studying PUFs that are publicly available (instead of the current standard of adding dedicated circuitry with PUF characteristics). This scenario provides new constraints and more challenging attacker models, due to the (public) accessibility of the PUFs. On the other hand it will also provide new possibilities, especially in reducing the cost and effort for manufacturers to include PUFs in their products.

a) Random Number Generation: Many cryptographic primitives rely on random data to ensure security. In particular the generation of keys, salts or nonces requires random data to be unpredictable to attackers. Therefore, Pseudo-Random Number Generators (PRNGs) need to be fed with high entropy seeds. The PUFFIN project investigates approaches which employ intrinsic hardware components as sources for such high entropy data like SRAM modules, GPUs or CPU cache states. PRNG solutions based on SRAM noise as in [3] could assure that cryptographically secure PRNGs can be deployed in contemporary consumer chips, mitigating security incidents due to weak RNG seeds (for example, see [4–6]).

b) Authentication: In contrast to present authentication approaches of using passwords or additional hardware dongles, PUFFIN is investigating approaches to utilize intrinsic PUFs as part of user's hardware as identifying token.

Two different schemes of authentication exist, namely 1.) client authentication and 2.) server authentication, both being based on a challenge-response scheme. While the verifier (server) possesses a database of challenge-response pairs (CRPs), linked to a defined PUF device and set up during an initial enrollment phase, the prover (client) is in the possession of the actual PUF device. In the more common client authentication scenario the prover needs to respond to a challenge, randomly chosen by the verifier, by querying its PUF and returning the measured response to the verifier. If the prover's response matches the expected one, the prover is authenticated. Server authentication is based on the same initial condition with the client being in possession of the PUF device and the server holding the CRP database. The prover, in this case the server, sends a randomly chosen CRP, which is verified by the client, by querying its PUF, to see whether the measured response matches the one provided by the prover.

Applying these protocols without further security measures, an adversary could capture exchanged CRPs and impersonate the victim, if a challenge is used more then once, since he now knows the corresponding response. Technically, this Man-in-the-Middle (MITM) attack can be solved by using PUFs with a large CRP space, i.e. an exponential number of CRPs. Thereby,

for each authentication request a new CRP is used, rendering MITM attacks impossible. One research topic for the PUFFIN project in this scenario is to prevent MITM attacks, given the fact that currently found intrinsic PUFs, in particular SRAM PUFs, exhibit a small CRP space, i.e. only a small number of CRPs. Using such PUFs without further protection enables the attacker to model the PUF by trying all possible CRPs in polynomial time. Therefore, our research focuses on the application of cryptographic protocols to mitigate the risk of MITM attacks and thus eliminate the possibility to model PUF behavior by capturing exchanged messages between server and client. Protocols like HBA^{++} [7] or the protocol from T. Dimitriou [8] are used for RFID authentication, where shared secrets are exchanged and security against MITM attacks is achieved. Amongst others, PUFFIN explores how to adapt principles of such protocols to the special requirements of intrinsic PUFs.

c) Device Identification: The use of PUFs for authentication purposes effectively turns the device into the authentication token. It supersedes the necessity to store a cryptographic key inside the device, which would be prone to attacks where an adversary tries to extract cryptographic material from non-volatile memory. Instead, the PUF device generates an ephemeral key 'on-the-fly' on the basis of it's unique physical characteristics, minimizing the attack surface to extract the key. However, the use of standard components as PUFs entails security-related challenges, which is a major subject of research to PUFFIN. For example, research is required on how to deal with a White-Box attacker (i.e. an adversary with access to the hardware and thus to the SRAM).

d) Secure Environment: The idea of PUF-based Secure Environment is based on the generation of keys depending on environmental features, which was introduced in [9]. In particular, the idea of this scenario is to generate a cryptographic key, which depends on the underlying hardware and thus implicitly identifies the device. Subsequently, the key is used to unlock encrypted software, which is installed on the device. More precisely, our goal is to decrypt the bootloader, which is executed first during device start-up in the domain of embedded devices. After the bootloader has been decrypted using the key, derived from the PUF response, it subsequently unlocks the kernel, which in turn decrypts user space applications. Since every layer relies on the preceding layer to be decrypted, it is possible to establish a chain-of-trust with the hardware constituting the anchor-of-trust.

A full implementation of the scheme could provide an alternative to current approaches to device identification. The TCG's Mobile Trusted Module (MTM) approach [10] for example relies on storing several keys and certificates in dedicated chips or in software. The former option requires additional hardware, which induces extra costs from the manufacturer's point of view. Alternatively, software MTMs can not provide strong hardware-based anchor-of-trusts per definition. PUFFIN strives to provide a low-cost hardware-based solution for mobile computing, overcoming both drawbacks. Furthermore, traditional approaches are potentially prone to side-channel attacks or other means to extract cryptographic material. In contrast, the proposed approach of using intrinsic PUFs would bind a software instance to the hardware itself and not to a permanently stored cryptographic key.

e) IP protection: By establishing a device identification scheme, which depends on the physical properties of the underlying hardware further use cases could be implemented as well. It would be possible to tightly intertwine user space applications with the underlying hardware, thus realizing an effective protection for intellectual property (IP protection) in terms of software. In particular, it could address the mitigation of illegal copies of software. Conceptually closely connected to IP protection is the approach of remote attestation, where the software configuration of a device can be securely transmitted to a third party. Since by using Hardware/Software-Binding a software instance is bound to a particular hardware, attestation of the software configuration is given implicitly.

III. Conclusion and Future Work

The PUFFIN project is studying the existence of (and use cases for) PUF instantiations that are readily available in consumer devices. Besides the challenge of actually finding these PUFs (e.g. in GPUs and CPUs), the project needs to deal with very specific constraints and attacker models. These specific requirements can either be due to the fact that the PUFs are publicly available or that most intrinsic PUFs only have a limited set of CRPs. If the project is able to find ways to overcome these challenges, PUFFIN will play a major role in reducing the cost and effort for manufacturers to include PUFs in their products. The first results of the project are very promising, but there still is a long way to go.

References

[1] Physically unclonable functions found in standard PC components (PUFFIN), "INFSO-ICT-284833," http://www.puffin.eu.org/.

[2] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, "FPGA Intrinsic PUFs and Their Use for IP Protection," in *Proceedings of CHES '07*, ser. LNCS, P. Paillier and I. Verbauwhede, Eds., vol. 4727. Berlin, Heidelberg: Springer-Verlag, 2007.

[3] V. van der Leest, E. van der Sluis, G. J. Schrijen, P. Tuyls, and H. Handschuh, "Efficient Implementation of True Random Number Generator Based on SRAM PUFs." in *Cryptography and Security*, ser. Lecture Notes in Computer Science, D. Naccache, Ed., vol. 6805. Springer, 2012.

[4] N. Heninger, Z. Durumeric, E. Wustrow, and J. A. Halderman, "Mining your Ps and Qs: Detection of Widespread Weak Keys in Network Devices," in *Proceedings of USENIX Security Symposium*, Aug. 2012.

[5] A. K. Lenstra, J. P. Hughes, M. Augier, J. W. Bos, T. Kleinjung, and C. Wachter, "Ron was wrong, Whit is right," Cryptology ePrint Archive, Report 2012/064, 2012.

[6] Debian Security, "DSA-1571-1 OpenSSL – Predictable Random Number Generator," Tech. Rep., May 2008. [Online]. Available: http://www.debian.org/security/2008/dsa-1571.en.html

[7] S. Piramuthu, "HB and Related Lightweight Authentication Protocols for Secure RFID Tag/Reader Authentication," in *CollECTeR*, 2006.

[8] T. Dimitriou, "A Lightweight RFID Protocol to Protect against Traceability and Cloning Attacks," in *SecureComm*, pp. 59–66.

[9] J. Riordan and B. Schneier, "Environmental Key Generation Towards Clueless Agents," in *Mobile Agents and Security*. London, UK, UK: Springer-Verlag, 1998. [Online]. Available: http://dl.acm.org/citation.cfm?id=648051.746194

[10] TCG, "Mobile Trusted Module Specification, Version 1.0, Revision 7.02," Trusted Computing Group, Tech. Rep., 2010.