Hardware Intrinsic Security to Protect Value in the Mobile Market

Vincent van der Leest · Roel Maes · Geert-Jan Schrijen · Pim Tuyls

Intrinsic-ID

{ vincent.van.der.leest | roel.maes | geert.jan.schrijen | pim.tuyls }
@intrinsic-id.com

Abstract

More and more mobile device manufacturers are recognizing the importance of security for their devices in order to protect valuable information of their customers. However, the security of many mobile devices currently does not suffice to protect against modern sophisticated attackers. This paper will go into detail on how these devices can be secured at the hardware level, to ensure that the data of mobile users can be protected against these skilled attackers. For strong protection anchored in hardware, this paper describes the concept of Hardware Intrinsic Security (HIS) and its security benefits for the mobile market. Using HIS technology a root of trust can be created in silicon, which is based on unique physical characteristics of the chips inside mobile devices. These characteristics can be thought of as the electronic fingerprint of a device, a technique also referred to as Physical Unclonable Functions (PUFs). A PUF is a basic building block for extracting this electronic fingerprint, but it does not provide a security solution by itself. In order to use a PUF in a security product or solution, it must be deployed in a controlled and secure way. This paper describes an extensively tested way of working for designing and implementing an electronic fingerprint, which is derived from these physical characteristics, into the systems required for providing strong security solutions on mobile devices.

1 Introduction

Only twenty years ago the Internet arrived on the PC. Today, even mobile devices and sensors are connected to the Internet of Things. As such, human beings are connected to each other as well as to machines all the time through the mobile devices they carry around. This has brought a lot of benefits in terms of economic productivity and ease of use. New ways of buying and paying, such as m-commerce and e-banking, have become available. However, through this (r)evolution our society has become completely dependent on electronic information exchange and storage on personal devices and network servers. Therefore the success and security of our society has become dependent on the adequate protection of all equipment involved, including the chips that can be found in mobile devices.

As long as semiconductor devices have been used to store, unlock or protect content of value they have been subjected to (attempts of) hacking. A rat race has been going on for several decades already between semiconductor manufacturers and motivated and skilled adversaries. In this race the adversaries attempt to break the security of chips in order to obtain access to the content that these devices are protecting. Given that mobile devices are storing more and more sensitive private and corporate data (e.g. documents, payments, e-mails), protecting these devices becomes increasingly more important. For this purpose more and more secure operating systems are being developed, like the Trusted Execution

Environment and TrustZone. However, in order for these environments to be secure, a "root of trust" in hardware is required. This root of trust is used to securely store the cryptographic keys that are required in the trusted operating system, e.g. to perform secure boot, data encryption and authentication. In this paper we will describe how to create such a root of trust in hardware of mobile devices based on Hardware Intrinsic Security (HIS) technology and what the advantages of this technology are in comparison to traditional key storage methods.

1.1 Threats to Semiconductor Security

Over the years, the number of attacks on semiconductor devices, and the sophistication of these attacks, has steadily increased. This clearly has an impact on security requirements. It turns out that, even for devices developed specifically for high security applications (e.g. smart cards), attackers manage to develop methods for opening the physical package of the chip and reading out security critical information from its non-volatile memories using electron microscopes and other advanced failure analysis techniques. Typical attacks on chips can be divided into three main categories:

- Side channel attacks (non-invasive, e.g. learning sensitive information from power or timing behavior).
- Fault attacks (semi-invasive, e.g. altering semiconductor behavior using laser light or ion beams and learning sensitive information from possibly faulty results).
- Invasive attacks (e.g. learning sensitive information directly from memories or reverse engineering of implementations).

From these physical attacks, the invasive attacks are clearly the most disruptive to security, because countermeasures on the physical level are required for protection from these kinds of attacks. Considering the progress made on these attacks, many standard countermeasures will not protect against future challenges. A recent example of how invasive attacks are a threat to semiconductor security can be found in the work of Tarnovsky [Flyl14]. He was, for example, able to break into Infineon's Trusted Platform Module SLE66 (designed for storing sensitive data), which was classified as "unhackable", as well as Atmel's ATMega2560.

In order to protect against these new attacks that will also threaten security in the mobile market, sensitive data and basic software code must be protected. Both volatile (SRAM) and non-volatile memories (ROM, EEPROM, Flash) on these chips are usually protected by memory encryption. If implemented properly, state of the art encryption algorithms are secure and therefore essential elements for protecting the content of any memory, volatile or non-volatile. These algorithms use a secret key, whose secure storage is consequently essential.

1.2 Traditional Key Storage Methods

For effective security, cryptographic keys need to be available within the device that requires them. Since external non-volatile key storage is vulnerable to simple eavesdropping, most common solutions rely on keys stored in on-chip non-volatile memory (NVM) or battery-backed SRAM. However, this approach entails certain possibly critical (security) issues:

- **Tampering:** NVM or battery-backed SRAM contents are always present, even when the overall system is not powered. As a result, they can be tampered with or read out using physical attacks based on techniques derived from IC failure analysis. Additionally, battery-backed SRAM presents a logistic problem for systems that have an extended lifetime requirement or a constrained form factor.
- **Programming:** Using this approach, the programming of secret keys in the field often relies on the IC/system manufacturer. This implies that a potentially untrusted third party is responsible for the initial secret key to enable key reprogramming. This opens

up a possible loophole that can be exploited to compromise keys within the product supply chain, outside of the customer's control.

- **Cost:** NVM comes at a substantial increase in area of a chip and cost of devices. Floating-gate-based NVM technologies require six to ten additional mask steps, which add significantly to the product cost. These, as well as anti-fuse-based techniques, also require a costly charge pump. Additionally, many NVM technologies have a potentially negative impact on yield. Battery-backed SRAM on the other hand has a significant space overhead on the PCB for placing the battery; space which is often not available.

1.3 Hardware Intrinsic Security

A new security approach that offers a clear advantage with respect to the issues identified above is Hardware Intrinsic Security (HIS). HIS technology uses the unique physical characteristics of a device (i.e., its electronic fingerprint) to derive a cryptographic key instead of storing it in NVM. In a device implementing HIS, the key bits are not present when the device is switched off. Furthermore, the system can be implemented such that keys are only present for a minimal amount of time in the device, which minimizes the window of attack.

These properties give HIS technology a security advantage over other key storage technologies where keys are permanently present inside the device. Also, the cost and overhead for integrating HIS technology is minor compared to most traditional key storage methods. HIS hence provides an improved security-cost trade-off for implementing a hardware root of trust, as depicted in Fig. 1.

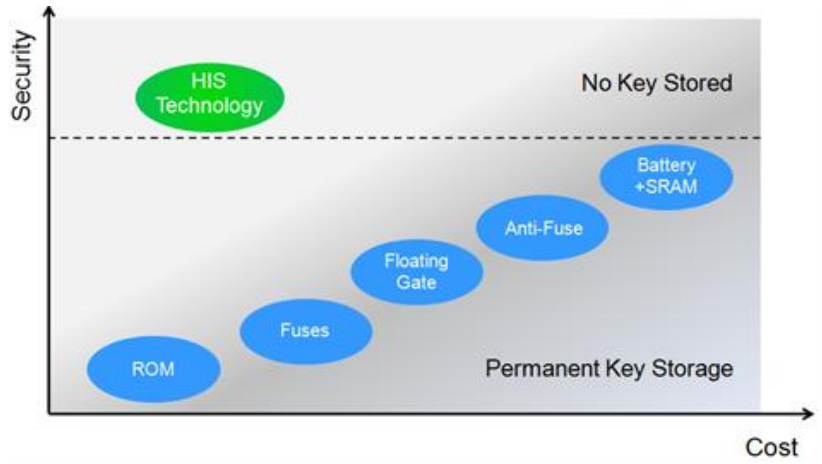

Fig. 1: Security-Cost positioning of various key storage technologies.

A lightweight method of storing keys in non-volatile memory is using **ROM**. However, this method offers virtually no physical protection of a key's secrecy and is moreover very inflexible when device-specific keys are required. This makes ROM an unfit choice for secure and flexible key storage.

Fuses are relatively big in silicon area and require a programming infrastructure (on-chip charge pumps or external infrastructure) which adds cost to the manufacturing process. **Floating gate** and **anti-fuse** storage technology have additional costs in the process since they use non-standard components. They require additional mask sets to be used during manufacturing and also the implementation of on-chip charge pumps.

SRAM on the other hand is a standard component that is available without additional costs. However, the battery that is needed for long-term storage in (volatile) SRAM is again a costly and bulky component.

Compared to these traditional key storage methods, HIS technology has a relatively low-cost implementation and a minimal integration overhead. Also, due to the properties that come along with deriving a key based on unique physical characteristics rather than storing a key in memory, HIS technology offers superior physical security compared to traditional methods and by its very nature provides storage of device-unique keys. In the next sections we will elaborate more on the principles of HIS and explain how a HIS-based system is built on top of Physical Unclonable Functions (PUFs).

2 Physical Unclonable Functions

Physical Unclonable Functions (PUFs) are known in the academic literature as electronic design components that derive device-unique properties, or electronic fingerprints, from integrated circuits (ICs). In modern deep submicron technology, the uncontrollable variations in feature dimensions and doping concentrations of silicon structures lead to a unique threshold voltage for each transistor on a chip. Since even the manufacturer cannot control these exact variations for a specific device, these physical properties are unclonable in practice.

Although it might be tempting to use the values of the threshold voltages directly as a unique identifier of the IC, their sensitivity to environmental conditions does not make this a viable option. There are various ways to implement PUFs in ICs that measure unique device properties in a more stable manner. They vary from comparing path delays and frequencies of free running oscillators to measuring startup data from memory components. Examples of papers describing PUFs include [GCDD02], [GKST07] and [MaTV08], an elaborate overview of PUF constructions is presented in [Maes13]. In this paper we focus on PUFs based on memory components since they have shown to be the most reliable and secure in practice, and hence economically viable. The main example of this PUF type is called an SRAM PUF, which is described in Section 2.2.

2.1 Reliability and Unpredictability

To qualify as a good PUF, a circuit element should possess two important properties: reliability and unpredictability. Reliability means that the variations in a specific PUF's measurements need to be sufficiently small over the lifetime of the IC, and over a wide range of external conditions e.g., temperature, voltage, electromagnetic fields, etc. So the device-specific electronic fingerprint is relatively stable, regardless of environmental conditions, during the lifetime of the IC.

Unpredictability, on the other hand, means that a random PUF's measurement needs to be unpredictably random under all circumstances i.e., the uncertainty about its value, or the PUF's entropy, needs to be sufficiently high. As a result of this property, it is also clear that each individual PUF circuit is highly unique. It can therefore produce its own characteristic electronic fingerprint that makes the IC uniquely identifiable.

PUFs used in HIS-based systems have been extensively tested and evaluated focusing on both of these properties.

2.2 SRAM PUF

A type of memory-based PUF that was proven to be most suitable for HIS-based products (see for example [KKRS12], [BhCM12]) is the so-called SRAM PUF, which was introduced in [GKST07]. SRAM PUFs are based on the power-up values of SRAM cells.

Every SRAM cell consists of two cross-coupled inverters. In a typical SRAM cell design, the inverters are designed to be nominally identical. However, due to the process variations during manufacturing, the electrical properties of the cross-coupled inverters will be slightly

out of balance. In particular the threshold voltages of the transistors in the inverters will show some random variation. This minor mismatch gives each SRAM cell a preference to power-up with either a logical 0 or a logical 1 on its output, which is determined by the stronger of the two inverters. Since this variation is random, on average 50% of the SRAM cells have 0 as their preferred startup state and 50% prefer 1.

We can evaluate the behavior of this SRAM PUF based on the two main properties for PUFs, reliability and unpredictability. Over the past years, thorough analysis of SRAM startup data has been performed. For this analysis startup patterns have been measured under various conditions, from SRAM implemented in several technology nodes (180nm down to 14nm) by several foundries with different processes. Extensive tests performed at Intrinsic-ID and their partners (e.g. in [KKRS12], [ScLe12]) have shown the following results:

- **Reliability:** The majority of the bit cells in an SRAM array have a strongly preferred startup value which remains static over time and under varying operational conditions. A minority of cells consist of inverters that are coincidentally well balanced and result in bit cells that will sometimes startup as a 0 and sometimes as a 1. This causes limited noise in consecutive SRAM startup measurements. Tests demonstrate that the noise level of the SRAM PUF under extensive environmental conditions (e.g. temperatures from -55˚C to 150˚C) and over years of lifetime is sufficiently low to extract cryptographic keys with overwhelming reliability using the appropriate HIS-based post-processing techniques.
- **Unpredictability:** Extensive testing demonstrates that the startup pattern of an SRAM array is unique for every IC and even for a specific memory within every IC. The startup bits are moreover highly uniform and completely independent of each other, making the pattern highly unpredictable and providing a large amount of entropy. The amount of entropy is sufficiently high to be able to extract secure and unique cryptographic keys.

Furthermore, the SRAM PUF has the advantage in comparison to other types of PUFs that it is based on a standard logic component that is available in all process nodes. No custom circuit needs to be designed or tuned. Combined with its very high reliability and unpredictability, this is an ideal PUF for use in HIS products.

2.3 Security of PUFs

It is important to stress that a PUF implementation by itself does ***not*** add much security to an integrated circuit. However, given their unique qualities, PUFs will serve as a physical security cornerstone for an IC design when integrated properly in a security system. Implementing PUFs in a secure manner and securely processing and using their outputs is not trivial and requires specific techniques. Expertise on how to do this is essential in raising the security level of a chip or system.

3 Hardware Intrinsic Security

The secure use of device-unique characteristics to strengthen the security of integrated circuits and systems is the focus of HIS. By extracting a cryptographic key from these characteristics, a highly secure alternative for traditional key storage methods in NVM is provided. As stated before, the implementation of a HIS-based system is built on top of an SRAM PUF. By using the random properties of the PUF's characteristics, both in the fingerprint itself and in the measurement uncertainty, unpredictable secret keys are generated and internal datapaths are protected.

3.1 Fuzzy Extractors

To extract a cryptographic key from the PUF's device-unique characteristics, the use of a so-called fuzzy extractor (also called key extractor or helper data algorithm, and conceptually shown in Fig. 2) is required. A fuzzy extractor, as introduced in [LiTu03] and [DoRS04], typically consists of two main parts:

- **An information reconciliation module** using error-correction techniques to deal with the noise in consecutive PUF responses. Public (non-sensitive) helper data provides the information needed to correct errors in the PUF response without revealing any information on the extracted key.
- **A privacy amplification module** using compression functions to extract the required entropy from the PUF response. This module guarantees that the output key is completely unpredictable and has full entropy, despite the availability of the helper data (i.e., even an attacker knowing the helper data can do no better than try every possible key combination).

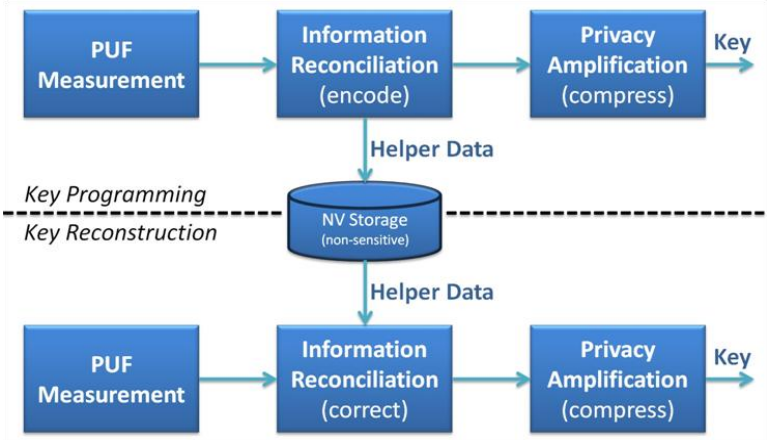

Fig. 2: Fuzzy Extractor architecture (programming and reconstructing keys)

3.2 Implementation of Fuzzy Extractors

In order to design and implement a good fuzzy extractor for a HIS-based system several aspects need to be taken into account. The PUF only provides the basic readout mechanism of an electronic fingerprint. A HIS-based product integrates this physical security cornerstone into a secure, reliable and economically attractive implementation of a security solution.

Security: Fuzzy extractor designs proposed in academic literature (e.g. [LiTu03], [DoRS04]) focus on theoretical security and performance aspects. However, there is a significant difference between such designs and a fuzzy extractor that is ready for practical hardware security products. Besides taking into account theoretical security issues, a product implementation requires resistance against practical attacks, and even needs to be prepared for attack methods that may arise in the future. Therefore, various attack countermeasures have been developed and implemented in HIS (see Section 3.3) to provide a layered, in-depth security approach.

Reliability: Reliability is an equally important design factor that is simultaneously taken into account. A fuzzy extractor in a HIS implementation is able to deal with noise on SRAM startup patterns caused by varying environmental conditions, like extreme and unforeseen usage circumstances, silicon ageing over product lifetimes of many years, and expected as well as unexpected operations (e.g., requirements for reset behavior, interrupt behavior, etc.).

Economic Aspects: When designing and implementing a fuzzy extractor, economic aspects also need to be considered:

- As with most silicon products, it is important for HIS-based products to have a small enough silicon footprint to make them commercially interesting. Silicon area translates directly into cost for the manufacturer and hence for their customers. The cost of the complete security implementation (including fuzzy extractor) must be as low as possible and in proportion with the overall value it is protecting. HIS provides a more secure solution with a smaller silicon footprint than traditional key storage solutions, which makes it better suited to serve multiple applications and markets.
- Other cost factors in the IC production process are the use of non-standard technologies, or the design of custom circuits, e.g. that require more masking steps in manufacturing, or additional test runs for circuit tuning. Therefore, HIS uses only standard logic components.

3.3 Security of HIS Products

When designing security systems it is important to take into account known attack methods as well as to protect preemptively against future attack vectors. Thorough analysis of known attacks on security systems leads to valuable insights in design principles and countermeasures that are important to thwart such attacks. Numerous design principles and countermeasures are included into the design to defend against known attacks on key storage and to minimize the success probability of yet unknown methods of attack. Examples of such countermeasures in HIS-based products are:

- Use a dedicated SRAM memory within a HIS implementation. In other words, the SRAM used for the HIS system is not accessible by any other process in the IC.
- Give the HIS module full control over the data access to the SRAM (i.e., the SRAM has no interface outside of the HIS system, which an attacker could exploit).
- Encrypt all data that is (temporarily) stored in the SRAM or in other internal registers.
- Let data processing logic detect anomalous behavior indicating ongoing attacks (e.g., bursts of highly repetitive operations, SRAM cells stuck at a particular bit value).
- Check proper functionality of the SRAM (PUF) and the fuzzy extractor using a Built-In Self Test (BIST).
- Protect the system from side channel attacks. Examples of countermeasures against side channel attacks include e.g., a randomized readout process of the SRAM, out-of-order operation, insertion of random delays, and parallelism of different processes.
- Integrate internal redundancy checks when processing data such that faults introduced by attackers cannot lead to leakage of sensitive information.
- Clear all internal data as soon as the device's attack sensors detect a possible attack.
- Switch off the SRAM's power whenever possible.

Combinations of these and other countermeasures are used in HIS products to provide a secure solution. Because of the countermeasures and design principles taken into account for HIS-based products, they are not vulnerable to known attacks on PUFs (e.g. [HBNS13], [NSHB13]), not even to invasive attacks through the backside of the silicon IC. Nonetheless, these academic attack proposals on PUF technology are continuously monitored and studied in detail to ensure HIS countermeasures are resistant and to strengthen the security of HIS products even further.

4 Use Case Example: Key Management Module

As stated before, for the security of mobile devices a root of trust in hardware is required. In this root of trust cryptographic keys are securely stored, so that a trusted operating system can be build on top of it. The keys stored in the root of trust are used for several important

cryptographic applications, e.g. secure boot, (a)symmetric crypto, authentication. Also, the use of multiple keys greatly strengthens key management procedures by enabling different levels of security access and functionality for different users and it allows applications to setup their own keys. Examples of how to build a secure environment on top of PUFs can be found in [ZZHQ14] (PUFs with ARM TrustZone) and [SALK14] (PUFs for secure boot).

In this section we will describe how this root of trust can be designed as a flexible key manager, based on the HIS approach. The key manager is able to store multiple keys for cryptographic purposes in a secure and efficient manner. This solution is based on an intrinsic Master Key (MK) that is unique for every device, never leaves the key management module and is used to wrap (encrypt and authenticate) all cryptographic keys before storing them.

4.1 Master Key Enrollment

The key management module based on HIS technology, as depicted in Fig. 3a, is prepared for operations by executing the one-time process of enrolling the Master Key (MK). During this phase, which is performed at the beginning of the device's lifetime, MK is derived based on a single readout of the SRAM startup pattern by the fuzzy extractor (equivalent to "Key Programming" in Fig. 2). This module also generates the helper data, which is non-sensitive data and can be stored in unprotected NVM without security risks.

4.2 Key Programming

Once MK has been derived, it is used to wrap the keys that need to be stored by the key management module. These keys can either be provided by the system or generated randomly (see Fig. 3a). Once these keys have been wrapped with MK, they can be stored in unprotected NVM (on- or off-chip) without security risks. Using this principle, which can be repeated whenever a new key is required, multiple keys are securely generated and stored by the key management module.

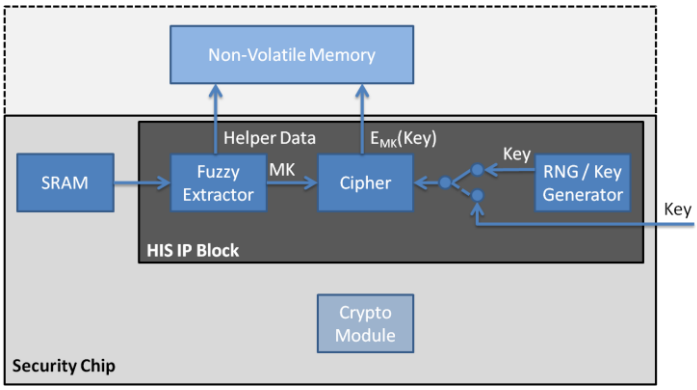

Fig. 3a: MK Enrollment and Key Programming

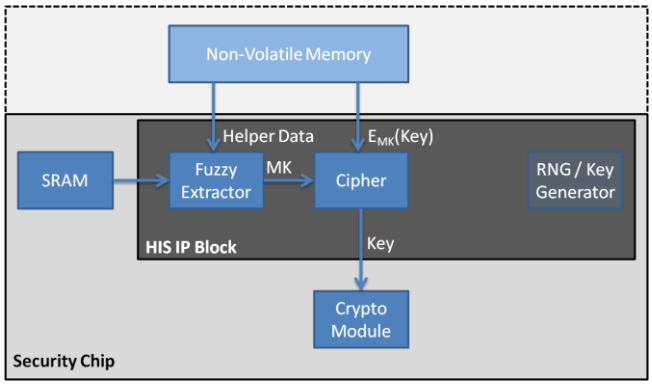

Fig. 3b: Key Reconstruction

4.3 Key Reconstruction

After programming the keys, the device is ready for Key Reconstruction. This phase (Fig. 3b) is enabled every time the device is powered-on after Key Programming and starts with the reconstruction of MK. This is done by combining the helper data (stored in NVM) with a new SRAM startup pattern when powering the memory. This new startup pattern is a noisy version of the pattern used during Key Programming. The fuzzy extractor is able to correct the noise on the SRAM startup pattern and reproduce MK at every power-up of the device (equivalent to "Key Reconstruction" in Fig. 2).

After MK has been reconstructed, it is used to unwrap the wrapped keys that have been stored in NVM. Once these keys have been decrypted and authenticated, they are used in the

designated crypto modules for which they are intended. Using this principle these keys can be reconstructed whenever they are required. They are also deleted from internal registers when they are not needed anymore. This way the window of opportunity for attackers to obtain these keys is minimized, since keys are only present when they are needed.

This section has shown how HIS technology is used to create a flexible management module, which stores a large number of cryptographic keys. Note that it is of vital importance to the security of this key management module that the design is compliant to the design methodologies of HIS technology. This system cannot be considered to be secure unless an appropriate set of the countermeasures described in Section 3.3 is implemented in the module.

5 Intrinsic-ID and HIS-based Products

Intrinsic-ID has developed several products based on HIS technology, including a secure and flexible key management solution (Quiddikey-Flex) and a full fledged root of trust solution for mobile devices (Confidentio-SC). HIS technology is used by Intrinsic-ID to solve security use cases for amongst others secure boot, hardware-software binding, and content protection (e.g. secure cloud storage) on mobile devices, but also on smart cards, microcontrollers and FPGAs. HIS technology has been extensively tested and approved by security experts at major entities, including Samsung, NXP, Microsemi, STMicroelectronics, Oberthur, Philips, Thales, SiVenture, and a major US defense contractor. More information about Intrinsic-ID, HIS technology and HIS-based products can be found at http://www.intrinsic-id.com.

6 Conclusion

A rat race has been going on for decades now in the semiconductor industry between manufacturers of secure ICs and hardware systems, and motivated adversaries aiming to break the claimed security. This ongoing competition is currently crossing over into the domain of mobile devices, and it has already been shown that many security measures currently in place in these devices do not suffice against sophisticated attackers with the means and experience to perform invasive attacks. Therefore, a strong root of trust in hardware is required to bootstrap trusted environments providing security for these devices and their applications.

This paper demonstrates how an efficient and highly secure key storage and management solution is accomplished through Hardware Intrinsic Security (HIS). HIS technology is used to solve many different security use cases for mobile devices, as it has already done in the past for smart cards, microcontrollers and FPGAs. HIS represents an innovative and system-wide design methodology founded on Physical Unclonable Functions or PUFs.

PUFs are innovative circuit elements but are on their own not sufficient to protect a system's security. Building a security product encompasses much more than simply adding functional blocks. Extreme care must be taken when designing such a product and integrating its components. Any addition to a system may introduce weaknesses if it is not done in a security-savvy way. HIS is based on years of research and development experience as well as extensive security and reliability testing. It encompasses amongst others design choices for PUFs, optimizations and protective measures for key storage and management, as well as system wide guidelines for implementing and integrating a HIS design into a product.

Intrinsic-ID's HIS products take PUFs to the next level in security. They overcome several weaknesses of traditional embedded key storage and management technologies, which make them a root of trust in hardware on top of which the security architecture of mobile devices is built. Strong protection against sophisticated attacks and successful evaluations by major entities show that HIS is suitable for protecting our most valuable private and corporate assets on mobile devices.

References

[Flyl14] Flylogic, “Flylogic Blog,” http://www.flylogic.net/blog.

[GCDD02] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, “Silicon physical random functions,” in ACM Conference on Computer and Communications Security (CCS’02). New York, NY, USA: ACM, 2002, pp.148–160.

[GKST07] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, “FPGA intrinsic PUFs and their use for IP protection,” in Workshop on Cryptographic Hardware and Embedded Systems (CHES ’07), ser. LNCS, vol. 4727, Berlin, Heidelberg: Springer-Verlag, 2007, pp. 63–80.

[Maes13] R. Maes, “Physically Unclonable Functions – Constructions, Properties and Applications”, Springer 2013, ISBN 978-3-642-41394-0, pp. 1-172

[MaTV08] R. Maes, P. Tuyls, and I. Verbauwhede, “Intrinsic PUFs from flip-flops on reconfigurable devices,” in Workshop on Information and System Security (WISSec 2008), Eindhoven, NL, 2008, p. 17.

[KKRS12] S. Katzenbeisser, U. Kocabas¸, V. Rozic, A.-R. Sadeghi, I. Verbauwhede, and C. Wachsmann, “PUFs: Myth, Fact or Busted? A Security Evaluation of Physically Unclonable Functions (PUFs) Cast in Silicon,” in Cryptographic Hardware and Embedded Systems (CHES) 2012, ser. Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2012, vol. 7428, pp. 283–301.

[BhCM12] M. Bhargava, C. Cakir, and K. Mai, “Comparison of bi-stable and delay-based Physical Unclonable Functions from measurements in 65nm bulk CMOS,” in Custom Integrated Circuits Conference (CICC), 2012 IEEE, 2012, pp. 1–4.

[ScLe12] G.-J. Schrijen and V. van der Leest, “Comparative analysis of SRAM memories used as PUF primitives,” in Design, Automation Test in Europe Conference Exhibition (DATE) 2012, march 2012, pp. 1319 –1324.

[LiTu03] J.-P. Linnartz and P. Tuyls, “New shielding functions to enhance privacy and prevent misuse of biometric templates,” in Audio- and Video- Based Biometric Person Authentication, ser. Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2003, vol. 2688, pp. 393–402.

[DoRS04] Y. Dodis, L. Reyzin, and A. Smith, “Fuzzy extractors: How to generate strong keys from biometrics and other noisy data,” in Advances in Cryptology - EUROCRYPT 2004, ser. Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2004, vol. 3027, pp. 523–540.

[HBNS13] C. Helfmeier, C. Boit, D. Nedospasov, and J.-P. Seifert, “Cloning physically unclonable functions,” in Hardware-Oriented Security and Trust (HOST), 2013 IEEE International Symposium on, 2013, pp. 1–6.

[NSHB13] D. Nedospasov, J.-P. Seifert, C. Helfmeier, and C. Boit, “Invasive PUF analysis,” in Fault Diagnosis and Tolerance in Cryptography (FDTC), 2013 Workshop on, 2013, pp. 30–38.

[ZZHQ14] S. Zhao, Q. Zhang, G. Hu, Y. Qin, and D. Feng, “Providing Root of Trust for ARM TrustZone using SRAM PUFs,” in Cryptology ePrint Archive: Report 2014/464, 2014. http://eprint.iacr.org/2014/464

[SALK14] A. Schaller, T. Arul, V. van der Leest, and S. Katzenbeisser, "Lightweight Anti-Counterfeiting Solution for Low-End Commodity Hardware Using Inherent PUFs ", to be published at 7th International Conference on Trust & Trustworthy Computing (TRUST) 2014.

Index

PUFs, HIS, hardware security, mobile security, root of trust, semiconductors