

# Hardware Intrinsic Security from Physically Unclonable Functions

Helena Handschuh, Geert-Jan Schrijen, and Pim Tuyls

## 1 Introduction

Counterfeiting of goods in general and of electronic goods in particular is a growing concern with a huge impact on the global economy, the society, and the security of its critical infrastructure. Various examples are known where companies suffer from economic and brand damage due to competition with counterfeit goods. In some cases the use of counterfeit components has even led to tragic accidents in which lives were lost. It has also recently become clear that counterfeit products can penetrate the critical and security infrastructure of our modern societies and hence cause a threat to national security. One of the difficulties to deal with this problem stems from the fact that counterfeit goods can originate from sources that are able to make copies that are very hard to distinguish from their legitimate counterpart. A first well-known aspect of counterfeiting is product cloning. A second much less known but increasingly dangerous aspect consists of overproduction of goods.

A special, but modern, case of counterfeiting is theft of Intellectual Property such as software and designs. The attractive part from the attackers' point of view is that it is relatively easy to steal and has a high value without having to do huge investments in research and development. From a high-level point of view one can state that the attack can be thwarted by using encryption and authentication techniques. Device configuration data or embedded software can, for example, be encrypted such that it will only run on the device possessing the correct cryptographic key. Since encrypted data is still easy to copy, it now becomes essential that the secret key is well protected against copying or cloning.

In order to deal with these two aspects of counterfeiting, a secret unclonable identifier is required together with strong cryptographic protocols. In this chapter we focus on a new way to address these problems: Hardware Intrinsic Security. It is based on the implementation and generation of secret physically unclonable identifiers used in conjunction with cryptographic techniques such as encryption and

---

H. Handschuh (✉)

Intrinsic-ID, San Jose, CA 95110, USA; ESAT-COSIC, Katholieke Universiteit Leuven, Leuven, Belgium

e-mail: [helena.handschuh@intrinsic-ID.com](mailto:helena.handschuh@intrinsic-ID.com)

authentication algorithms which allow to secure the critical information stored in the system. According to common practice in security, the used algorithms are often public but they use a secret key that is stored securely *somewhere* in the system. Using secret physically unclonable identifiers to derive secret keys for the system is our proposed solution to achieve strong anti-counterfeiting and anti-cloning mechanisms in electronic devices.

In every security system, it is essential that the key remains completely secret to keep a high level of protection. The system is broken, i.e. does not guarantee protection anymore when the secret key has leaked. Nowadays, encrypted texts created with state-of-the-art cryptographic algorithms do not leak much information on the secret key. However, since secret keys are stored in everyday objects like smart cards, attackers can easily subject such objects to physical attacks with all kinds of tools in order to get access to the secret keys. Common examples of such tools are very high-resolution microscopes such as optical, atomic force, scanning electron, laser scanning, confocal microscopes, or more destructive tools such as focused ion beams and laser cutters. It has been shown in many occasions that by using these physical means the secret key bits can be visualized and hence the secret key can be retrieved. Although these tools are sophisticated, they are more and more widespread nowadays and affordable for many parties.

Currently an arms race between security IC manufacturers and attackers is taking place to protect the secret keys in improved ways. It turns out, however, that the traditional methods to protect secret keys are approaching their limits and inducing more and more costs and longer time to market. A low cost but strong secret key storage technology is one of the missing links to make affordable but strong security systems. It is a necessary requirement for ICs in smart cards, defense and governmental applications, e-health systems, passports, and so on that protect valuable and sensitive data and that upon failure would cause not only very huge financial losses but also brand and reputation damage and could even expose a nation's critical assets.

Secure key storage is a small but indispensable part of a security system. Since a security system is only as strong as its weakest link, it is important to have a strong key storage mechanism. Moreover, when a secure and *unclonable* key storage mechanism is combined with good cryptography, a strong anti-counterfeiting system can be built. The unclonable key is used as a unique identifier and transfers its unclonability to the product it is embedded in. In order to detect whether a product has been counterfeited, a so-called *authenticity check* is performed. The authenticity check is usually carried out in a protocol between a verifier and the component to be verified. For example, the protocol could be run between a reader and an unclonable smart card or RFID Tag, or between a program running on a processor and the unclonable chip that implements the processor. In the first example, the unclonability of the device guarantees that when the verification succeeds the device is genuine. In the second example, the verifier is embedded in the program. It will authenticate the IC by verification of its secret key in a secure protocol. As a result, the program will not run on a counterfeit IC and protect the Intellectual Property contained in the processor design.

## 2 Rethinking Secure Key Storage Mechanisms

Current key storage mechanisms produce secret keys that are stored on the device that carries out the security operations. Off-chip storage of a secret key is vulnerable to a competent attacker using a logic analyzer to tap the bus between the external memory and the chip.<sup>1</sup> Therefore the storage mechanisms below have to be considered as embedded on-chip storage systems.

### 2.1 Limitations of Current Key Storage Mechanisms

A number of approaches exist to permanently store keys in a device. Among these we distinguish between volatile and non-volatile approaches. Non-volatile mechanisms rely on hardwired information or fuse-type technologies or floating gate-type technologies. Volatile approaches based on RAM memory typically use batteries to permanently store information. In this section we provide an overview of the limitations of each type of permanent storage mechanism before highlighting the advantages of our new proposed solution.

- *ROM memory.* ROM (read-only memory) masks are typically generated during manufacturing stages and can thereafter not be erased or modified anymore. This has two implications. First of all, any secret key hidden in ROM is permanently stored there even if the device is powered off and can therefore be extracted with typical failure analysis tools used at manufacturing sites. Second, ROM is about as inflexible as carving the key in stone. Once it has been designed into the IC and taped-out it can never be changed again. In terms of time to market, ROM masks take a number of months to be produced. Since it is impossible to consider that every new device would receive a new key and require a new ROM mask, this implies that ROM stored keys are necessarily master keys and all the more interesting to reverse engineer.
- *Fuse-based storage mechanisms.* Examples of fuse-based storage mechanisms are polyfuses, laser fuses, e-fuses, and anti-fuses. Again, as is the case with ROM memory, the keys stored in these fuses are permanently present in the system even when the device is powered off. Additionally fuses are quite easy to spot in a lay-out because they are quite large; they are all the easier to analyze using typical failure analysis tools from manufacturing sites. Some types of fuses, namely anti-fuses, require an additional charge pump in the system and are thus not as cost-efficient as one might hope for.
- *Floating gate technologies.* These technologies include Flash memory, EEPROM, and EPROM cells. The principle is that an electronic charge is trapped on the floating gate between two drains and remains there until a given threshold voltage is applied to remove it. Again, the information is trapped in the device

---

<sup>1</sup> Note that in systems where the external memory is encrypted, there still needs to be an on-chip key to decrypt the data from the memory as it is being read or written.

even when it is powered off and can be read using advanced imaging and failure analysis tools. Floating gate technologies are also vulnerable to fault attacks in which one tries to erase or modify the value trapped on the floating gate while being read or written and infer secret information from the consequences of the modification. Floating gate memory technologies are by no means standard technology components and appear only as process options in the later generations of a new process node. For a customer requiring a new technology node this can cause a substantial delay in the time to market of the product. Floating gate-based technologies also need 6–10 additional mask steps which adds significantly to the product cost. Due to the complicated nature of the processes for these various non-volatile technologies, it is at this point in time believed that it is not economically viable to have all these technologies available in all the process nodes. For example, embedded Flash is only available down to 90 nm technology at this time.

- *Battery-backed RAM.* Battery-backed RAM does not suffer from the security issues most other storage mechanisms have, but has one clear disadvantage compared to all others: It requires an additional component, namely a battery. This induces additional cost and assumes that there is enough room in the system to add a battery. In most embedded ICs, this is not the case. Another drawback of batteries is that they are not always very reliable and the information in the RAM is lost if they fail. This means that such devices can easily become nonfunctional.

As can be seen from the previous discussion, every current key storage mechanism has a number of limitations which cannot be easily overcome, the main one certainly being the permanent presence of the key in the system even when it is powered off.

## ***2.2 A Radical New Approach to Secure Key Storage***

Given the drawbacks of the current non-volatile storage mechanisms as described above, there is clearly an exposed gap in hardware security which is playing into the hands of determined attackers. To counter this increasing threat a radically new approach to key storage is needed. Important criteria for this new approach are the following:

1. First of all, the key should not be permanently stored in digital form on the device.
2. Second, it should be extracted from the device only when required. And after having been used, it should be removed from all internal registers, memories, and locations so as to not leave a single trace when the system is powered off again.
3. Third, it should somehow be uniquely linked to a given device such that one cannot reproduce it or manufacture a device with a precise key.

Our new approach that extracts the key from the intrinsic properties of the device overcomes many of the limitations of traditional approaches mentioned above. The implementation of such an approach without the need for technology-dependent components or embedded non-volatile memory has the following advantages:

- *Security*: It offers an unparalleled security level since the key is not even present when the device is switched off. It can be seen as key storage without storing the key.
- *Cost*: It does not require any additional mask steps or additional analog components. Therefore this solution saves cost instead of adding costs as compared to key storage alternatives.
- *Time to Market*: It is ready to use with the newest process nodes without requiring the extensive qualifications required for new process options.
- *Standard Availability*: Clearly, when properties of standard components are used to extract the key, the solution is available in most common process nodes.
- *Flexibility*: It is field upgradeable. Keys based on this principle can be updated in the field even after the device has left the production facility.
- *Reliability*: It offers reliability against a wide range of external influences, such as temperature and voltage variations and humidity. It does not suffer from the presence of an additional component such as a battery and remains stable throughout the device's lifetime without really being there.

### 3 Hardware Intrinsic Security

#### 3.1 Physically Unclonable Functions

The concept of a physical unclonable function (PUF) forms the basic idea on which the implementation of our new key storage approach is built. PUFs will be used as the hardware from which the key is extracted and can be considered as the intrinsic electronic fingerprint or biometric of a device. We will refer to security mechanisms built on electronic fingerprints as *Hardware Intrinsic Security*. The underlying electronic PUF technology has been extensively investigated in the literature and has been recognized as a new powerful security primitive. The previous chapter in this book by Maes and Verbauwhede provides a complete overview of existing PUF technologies and their essential properties.

An electronic PUF consists of a physical object that is very hard to clone due to its unique micro- or nano-scale properties that originate from the (deep-submicron) manufacturing process variations. An electronic PUF has to meet the following requirements:

1. *Low Cost*: The measurement circuit should be low cost and easy to implement, i.e. with standard components.
2. *Resistance to Physical Attack*: A physical attack meant to find out the behavior of the structure should cause damage to the structure. In particular this implies

that the functional behavior of the PUF should change to such an extent that tampering is detectable. The PUF should not be based on a secret that has to be guarded securely.

3. *Reliable*: The PUF responses should exhibit a low amount of noise in a wide range of circumstances, e.g., when being present in low- and high-temperature environments, environments with electromagnetic radiation, or environments that cause changes in the operating voltage of the device. Finally, still after many years of silicon aging effects, the noise level should be sufficiently low. The next chapter in this book by Schaumont et al. provides a thorough analysis of such aging effects.

### 3.1.1 Unclonability

PUFs are by definition very hard to clone. This means that it is very difficult, i.e., takes a lot of resources and a lot of time to make either a hardware clone, a mathematical model of the behavior of the structure, or a software program that can compute the response to a challenge in a reasonable amount of time. In order to be able to perform these actions, one would have to know the locations and properties of all the particles in the system with very high accuracy. Since physical systems consist of a very large amount of particles, this becomes a very time-consuming task.

### 3.1.2 Biometrics

There is a striking analogy between intrinsic PUFs and biometrics, in fact an intrinsic PUF can be seen as the biometric modality, i.e., the intrinsic electronic fingerprint of an IC. Even the ways of working with PUFs and biometrics are very similar. Both require a registration phase: it is necessary to perform some pre-processing before one can work with them. Once the pre-processing has been performed and some reference data based on this has been stored, the biometric/electronic fingerprint can be used for authentication and key storage purposes.

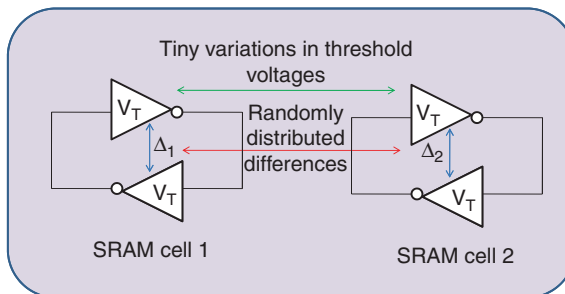
## 3.2 Examples of PUFs

### 3.2.1 SRAM PUFs

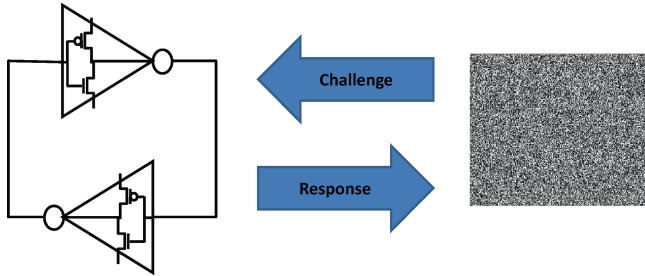
The best known memory-based intrinsic PUF based on standard available components is the SRAM PUF. Other memory-based intrinsic PUFs are also described in the previous chapter by Maes and Verbauwhede. SRAM or static random access memory is a standard component that is used in most devices (e.g., ASICs, microprocessors, DSPs, ASSPs) today. It consists of two cross-coupled invertors and two additional transistors for external connection, hence six transistors in total. It is widely used due to its speed for short-term data storage.

When a voltage is applied to a memory cell, it chooses its logical preference state: the logical 1-state or the logical 0-state. Each cell has a unique preference state due to its composition; the composition determines the values of the threshold voltages in the transistors that make up the two cross-coupled invertors. The unique properties of each transistor stem from deep submicron process variations. It is known that the fluctuations in the threshold voltages scale according to the law of Pelgrom:  $\Delta(V_T) \sim \frac{1}{\sqrt{LW}}$  where  $L$  is the length and  $W$  is the width. A complex interaction between all these physical variables determines in the end the logical preference states of the memory cells. The important observation in this example is that the threshold voltages of different transistors may well seem almost identical at the macroscopic level but that it is the difference between two of these threshold voltages that will actually govern the start-up value of each individual cell. Due to tiny local process variations, it is the difference between these differences that leads to a completely random start-up behavior of neighboring SRAM cells on a device as shown in Fig. 1.

The string determined by all the preference start-up values of the memory cells of an SRAM memory array forms a random identifier that identifies the SRAM memory uniquely. This identifier is the PUF response. A schematic representation of the SRAM PUF is shown in Fig. 2. This phenomenon has been verified in many experiments and on many SRAM types. Among the devices we have tested in our own facilities we can list the following: Alliance SRAMs, Cypress SRAMs, IDT SRAM, Faraday Standard Performance SRAM, and Virage Logic SRAMs, both High-Density and High-Speed. All these SRAM memories cover a large range of technology nodes, namely 180, 150, 130, 90, and 65 nm from different foundries, namely UMC and TSMC. A number of experiments were performed for each and every one of them showing that such SRAM memories do indeed start-up in a random fashion and are suitable for PUFs over a large range of environmental conditions.



**Fig. 1** Schematic representation of the differences in threshold voltages between two neighboring SRAM cells. Even though the threshold voltages are almost identical, their tiny differences are randomly distributed



**Fig. 2** Schematic representation of an SRAM PUF. The *left side* represents a single SRAM cell, consisting of two cross-coupled inverters. On the *right-hand side*, the SRAM PUF response of a whole SRAM memory is shown, where a *black pixel* can be interpreted as a logical 1 and a *white* as a logical 0

### 3.2.2 PUFs on FPGAs

Since the SRAM PUF is not available on all mainstream FPGA platforms (because no uninitialized SRAM is available on most types) we present briefly two examples of other types of PUFs that are targeted toward and can be configured on FPGAs: (i) the Butterfly PUF and (ii) the Ring Oscillator-based PUF. These two as well as further examples of memory-based and delay-based electronic PUFs are introduced in the previous chapter of this book by Maes and Verbauwhede.

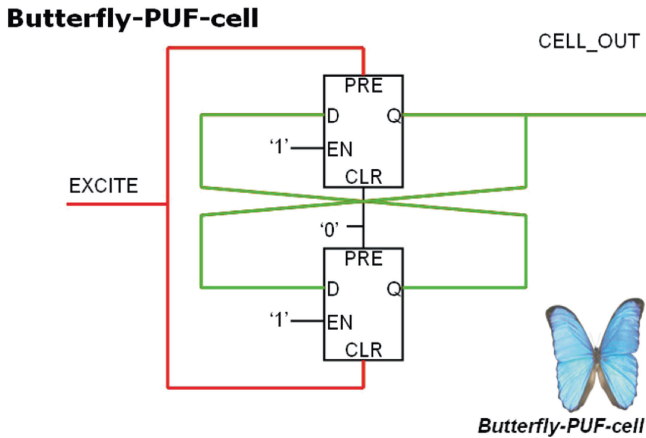
#### Butterfly PUF:

The idea behind the Butterfly PUF is similar to the one behind the SRAM PUF. At a high level it consists of two integrated components: (i) an array of Butterfly PUF cells and (ii) a processing component. A single Butterfly PUF cell consists of two cross-coupled latches. Due to this cross-coupling the Butterfly cell has two stable states the logical “0” and the logical “1,” just as the SRAM PUF cell. The cell is challenged by bringing this system into an unstable state and letting it converge during a specific time interval to one of the stable states. The preferential stable state is determined by the mismatch defined by the process variations during manufacturing. Its stability with respect to external stresses is guaranteed by tight integration with the processing component. In Fig. 3, a schematic overview of a Butterfly PUF cell is shown.

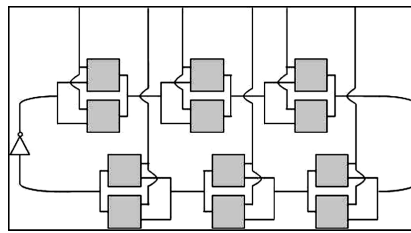
#### Ring Oscillator-based PUF:

This PUF consists of an oscillating loop that is constructed by putting a number of delay elements next to each other and feeding the signal back to its starting point. The frequency at which this circuit oscillates is determined by the physical properties of its building blocks and can therefore be used as a basis for a unique identifier. By measuring the unique oscillating frequencies of a number of these loops, a unique identifier is generated that can be translated into a secret key. Since the PUF





**Fig. 3** Schematic representation of a single Butterfly PUF cell on an FPGA



**Fig. 4** Schematic representation of one oscillation loop of a ring-oscillator PUF

responses are analog values, the PUF is integrated with a secure and noise-reducing analog to digital conversion algorithm. Apart from security, it guarantees robustness of the PUF responses against external stresses. In Fig. 4 a schematic representation of one oscillation loop of the Ring Oscillator PUF is presented.

### 3.3 Secure Key Storage Based on PUFs

Our proposed method of deriving the key using a PUF comprises two stages:

- *Noise Cancellation:* Physical measurements are typically noisy. Secret keys used in the context of cryptographic algorithms must always be exactly the same. Otherwise they produce completely corrupt results. Consequently, noise has to be removed from the physical measurements before they can be used to create secret keys.
- *Randomness Extraction:* Even after noise has been removed, a further processing step is required. The security from the cryptographic keys is based on the fact that they are completely random from one device to the next, i.e., very hard to guess. Physical measurements have a high degree of randomness but are usually not uniformly random. By processing the physical data and extracting the randomness

via appropriate compression functions (extractors), a uniformly random key can be generated.

For a practical implementation of such a key derivation mechanism, in order to use, for example, an SRAM PUF for Hardware Intrinsic Security, three functional modules are needed:

- *A PUF Measurement Circuit*: A measurement circuit that is able to read out the device-unique characteristics of the PUF and translate this into digital *PUF Data*. In case of an SRAM PUF, this is simply a circuit that reads out the start-up values of a specific range of SRAM memory that is exclusively reserved for this purpose.
- *A Key Extractor*: This is a module that converts noisy PUF responses into a robust secret key. It implements the noise cancellation and randomness extraction algorithms. Besides the PUF responses it needs an *activation code* as input. This activation code contains error correction data needed to remove the noise from the PUF data and information about the compression function needed to extract randomness. The Key Extractor module can be implemented not only as an IP block integrated in an IC but also as a software module that runs on an (embedded) processor.
- *An Activation Code Constructor*: This module computes the public activation code that is needed by the Key Extractor. It takes as input the PUF data and optionally a user-selected key that needs to be reconstructed in the future. The module can be implemented as an IP block on the same IC as the key extractor is located or as part of an external device or service, depending on the application.

Typically the Activation Code Constructor is used only once in a so-called *enrollment* phase. Once the Activation Code is generated, it is stored in a memory that is accessible by the Key Extractor. Note that this memory may be external to the device on which the Key Extractor is implemented and does not need to be secure. Each time the device needs to use the secret key, a new PUF measurement is done and the Key Extractor is used to reconstruct the key from the measured PUF data and the stored Activation Code. This is called the *reconstruction* phase. The reconstruction phase is typically carried out many times during the lifetime of the device (each time the key is needed).

## 4 Quality of a PUF

The quality of a PUF is determined by two main parameters which are reliability and security. Reliability addresses the fact that a PUF has to work under many different external circumstances and has to have a sufficiently long lifetime. On the other hand, security addresses the level of protection offered against a wide range of attacks. A very important parameter for security is the amount of randomness or entropy present in the PUF. A further in-depth discussion of reliability and statistical modeling of PUFs is performed in the next chapter of this book by Schaumont et al.

## 4.1 Reliability

Electronic PUFs are based on features of electronic components whose behavior under varying operating conditions is modeled and tested extensively before a PUF is commercially deployed. It must be guaranteed that the cryptographic key or unique identifier derived from the PUF is exactly the same under all circumstances. The following operating conditions can have an influence on the PUF behavior:

- Temperature
- Core voltage
- Electromagnetic radiation.

The influence of these conditions has been investigated by continuously reading out data from PUF implementations while varying the above-mentioned conditions in a climate chamber. The tests we performed included the following: measuring PUF responses under different ambient temperatures and under a gradient of temperatures, typically ranging from  $-40^{\circ}\text{C}$  to  $+80^{\circ}\text{C}$ ; measuring the PUF responses at extremely low and extremely high temperatures, sometimes up to  $125^{\circ}\text{C}$ , and at very high humidity levels; measuring the PUF responses at different core voltage levels; measuring the PUF responses when exposed to different electromagnetic fields. The differences between the measured PUF data and reference measurements taken in a controlled environment were analyzed. It turns out that PUFs are very robust with respect to these variations for a wide range of SRAM types as well as FPGA devices and families of FPGA devices. For some PUF types some data processing is used to make a particular implementation robust against such influences.

Besides dealing with a variety of operating conditions, it is also important to guarantee that a PUF works properly over time. It is known that silicon slowly degrades when in use for a long time. Several mechanisms contribute to this aging effect, the most important ones being

- Electro Migration (EM): the transport of conductor material due to momentum exchange between electrons and the metal lattice.
- Hot Carrier Injection (HCI): carriers generate sufficient kinetic energy to overcome a potential barrier and get injected into the gate oxide, causing interface states and charge traps.
- Time-Dependent Dielectric Breakdown (TDDB): formation of conducting path through the gate oxide.
- Negative Bias Temperature Instability (NBTI): build up of interface charges due to a negative gate-source bias at an elevated temperature.

These mechanisms can influence the behavior of the PUF over time. Depending on the type of PUF, different mechanisms are of importance. For example, the most important aging effects for oscillator-based PUFs are NBTI and HCI. FPGAs incorporating this technology have been submitted to extensive stress tests simulating the aging effect due to both NBTI and HCI. The result is that aging effects have almost no influence on the behavior of an oscillator PUF. As a matter of fact, none of the tests that we performed on the FPGAs both under extreme operating conditions and

simulating aging effects ever resulted in the cryptographic keys being wrong. The same key was always reconstructed no matter how the devices were stressed. The most important aging effect for the SRAM PUF is NBTI. In order to investigate its influence, experiments were done where the NBTI effect was accelerated by applying an increased voltage on the SRAM memory and by placing it in an environment with a high ambient temperature for a long time. This way an effective aging of 10–20 years was achieved in only a few months time. The experiments showed that if no countermeasures are taken, the start-up behavior of the SRAM PUF is changing. However, when the right countermeasure (or anti-aging mechanism) is applied, the impact of aging vanishes completely and even the noise on the derived PUF data is reduced.

4.2 Security

Three important security parameters of a PUF are its entropy, its tamper evidence and its unclonability. These properties are discussed below.

4.2.1 Entropy

In order to extract a high-quality secret key from a PUF, a sufficient amount of randomness is needed in the PUF responses. In the literature the amount of entropy present in various PUFs was analyzed. An overview is given in Table 1.

4.2.2 Tamper Evidence

PUFs provide very strong protection against physical attacks and are therefore very well suited to implement *read-proof hardware*. Read-proof hardware is hardware that is very hard to read by an attacker even when a whole arsenal of physical tools is available. Hence, a good key storage mechanism should be implemented by read-proof hardware.

Physical attacks can be *invasive* as well as *non-invasive*. An invasive physical attack is defined as an attack where the attacker physically breaks into a device and thereby modifies its structure. A non-invasive physical attack is one where the attacker performs measurements without modifications to the device’s structure.

When a PUF is attacked in a physical manner, its behavior will change. By this we mean that when the same challenge is applied to a PUF, a substantially different response will be generated. A substantially different response is a response whose noise level (w.r.t. to an enrollment measurement) is higher than the noise

Table 1 Entropy of different PUF types	
PUF type	Entropy per 1,000 bits
SRAM PUF	950
Delay PUF	130
Butterfly PUF	600

level of responses caused by environmental stresses. The implementation of a detection mechanism of these higher noise levels, allows the device to take appropriate measures when an attack is detected. In case when the PUF responses are used to implement a secure key storage mechanism these higher noise levels lead to a substantially different secret key being generated. Effectively this implies that the secret key in the device is being destroyed and cannot be recovered by an attacker anymore.

In order to assess the security of PUFs against invasive attacks, we submitted our SRAMs to an independent evaluation facility. This lab concluded that the most efficient way to attack these SRAMs consisted of trying to apply voltage contrast attacks. After experimenting for some time and trying different delayering techniques, it turned out that either the chips were functionally destroyed and could not operate anymore or no voltage contrast could be seen on the SRAMs for those which were still functional. This is mainly due to the fact that successful voltage contrast attacks require a very high voltage to be applied to the device and most devices simply do not survive such experiments. As a consequence, SRAM PUFs are shown to be resistant against voltage contrast attacks; the results of voltage contrast attacks on SRAMs will be described in a future detailed publication. We conclude that SRAM PUFs indeed qualify as a read-proof hardware implementation.

### 4.2.3 Unclonability

The fact that PUFs are unclonable implies that they can be used for anti-counterfeiting purposes and secure key storage.

When PUFs are used for the detection of the authenticity of a product, a physical property of the PUF is measured, translated into a bit string and verified. The physical unclonability of PUFs prevents building of a similar physical structure that upon interrogation produces a similar bitstring that would pass the verification test as the original one.

When the PUF responses are used as a source for secret keys, it is important that the PUF responses are only dealt with within the device to keep them protected from the attackers. In that way, one is protected against attackers that would be able to make a literal clone from a design point of view. Since clones based on an identical design do not translate into literal physical clones, the attacked devices will not have the same secret key or identifier as the original one.

## 5 Conclusions

In summary, a radically new approach, *hardware intrinsic security*, is available today to prevent cloning of semiconductor products and preserve the revenues of those companies. PUFs are used to generate the intrinsic fingerprint inherent in each device which is combined with a unique activation code to produce the secret key. No key is actually stored in hardware thereby significantly raising the level of security available beyond alternative methods.

## References

1. F. Armknecht, R. Maes, A.R. Sadeghi, B. Sunar, P. Tuyls, in *Memory Leakage-Resilient Encryption Based on Physically Unclonable Functions*. Proceedings of ASIACRYPT 2009. Lecture Notes in Computer Science, vol 5912, (Springer, 2009) pp. 685–702
2. C. Bösch, J. Guajardo, A.-R. Sadeghi, J. Shokrollahi, P. Tuyls, in *Efficient Helper Data Key Extractor on FPGAs*. Proceedings of CHES 2008. Lecture Notes in Computer Science, vol. 5154 (Springer, Heidelberg, 2008), pp. 181–197
3. B. Gassend, D.E. Clarke, M. van Dijk, S. Devadas, in *Controlled Physical Random Functions*. Proceedings of the 18th Annual Computer Security Applications Conference (ACSAC 2002), Las Vegas, NV, USA, 9–13 Dec 2002 (IEEE Computer Society, Washington, DC, 2002), pp. 149–160
4. M.A. Gora, A. Maiti, P. Schaumont, in *A Flexible Design Flow for Software IP Binding in Commodity FPGA*. IEEE Fourth International Symposium on Industrial Embedded Systems - SIES 2009, Ecole Polytechnique Federale de Lausanne, Switzerland, 8–10 July 2009 (IEEE, Piscataway, NJ, 2009) pp. 211–218
5. J. Guajardo, S.S. Kumar, G.-J. Schrijen, P. Tuyls, FPGA intrinsic PUFs and their use for IP protection, in *Cryptographic Hardware and Embedded Systems - CHES 2007*, ed. by P. Paillier, I. Verbauwhede. Proceedings of the 9th International Workshop on Cryptographic Hardware and Embedded Systems, Vienna, Austria, 10–13 Sept 2007. Lecture Notes in Computer Science, vol. 4727, (Springer, Berlin, Heidelberg, 2007), pp. 63–80
6. J. Guajardo, S.S. Kumar, G.J. Schrijen, P. Tuyls, in *Brand and IP Protection with Physical Unclonable Functions*. IEEE International Symposium on Circuits and Systems - ISCAS 2008, 18–21 May (IEEE, Piscataway, NJ 2008), pp. 3186–3189
7. J. Guajardo, B. Skoric, S.S. Kumar, T. Bel, A.H.M. Blom, G.J. Schrijen, Anti-counterfeiting, key distribution and key storage in an ambient world via physical unclonable functions. *Inf. Syst. Front.* **11**(1), 19–41 (2009).
8. T. Jun, *Circuit Approaches to Physical Cryptography*. Diploma thesis, Technische Universität München, 2009
9. S.S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, P. Tuyls, in *The Butterfly PUF: Protecting IP on Every FPGA*, ed. by M. Tehranipoor, J. Plusquellic. Proceedings of the IEEE International Workshop on Hardware-Oriented Security and Trust, HOST 2008, Anaheim, CA, USA, 9 June 2008 (IEEE Computer Society, Washington, DC, 2008), pp. 67–70
10. K. Kursawe, A.R. Sadeghi, D. Schellekens, B. Skoric, P. Tuyls, in *Reconfigurable Physical Unclonable Functions - Enabling Technology for Tamper-Resistant Storage*. 2nd IEEE International Workshop on Hardware-Oriented Security and Trust (HOST 2009), San Francisco CA, USA, 27 July 2009 (IEEE, Piscataway, NJ, 2009), pp. 22–29
11. J.W. Lee, D. Lim, B. Gassend, G.E. Suh, M. van Dijk, S. Devadas, in *A Technique to Build a Secret Key in Integrated Circuits for Identification and Authentication Applications*. Proceedings of the IEEE VLSI Circuits Symposium, Honolulu, HI, June 2004, pp. 176–179
12. D. Lim, *Extracting Secret Keys from Integrated Circuits*. Master thesis, Massachusetts Institute of Technology, May 2004
13. R. Maes, P. Tuyls, I. Verbauwhede, in *Intrinsic PUFs from Flip-Flops on Reconfigurable Devices*. 3rd Benelux Workshop on Information and System Security (WISec 2008) Eindhoven, The Netherlands, 13–14 Nov 2008 17 pages
14. R. Maes, P. Tuyls, I. Verbauwhede, in *Low-Overhead Implementation of a Soft Decision Helper Data Algorithm for SRAM PUFs*. Proceedings of CHES 2009, Lecture Notes in Computer Science, vol. 5747 (Springer, Berlin, Heidelberg, 2009), pp. 332–347
15. C.W. O'Donnell, G.E. Suh, S. Devadas, *PUF-Based Random Number Generation*, CSAIL CSG Technical Memo 481, Massachusetts Institute of Technology, Mar 2001
16. R.S. Pappu, *Physical One-Way Functions*. Ph.D. thesis, Massachusetts Institute of Technology, Mar 2001

17. B. Skoric, G.-J. Schrijen, W. Ophey, R. Wolters, N. Verhaegh, J. van Geloven, Experimental hardware for coating PUFs and optical PUFs, in *Security with Noisy Data - On Private Biometrics, Secure Key Storage and Anti-Counterfeiting*, ed. by P. Tuyls, B. Skoric, T. Kevenaar (Springer, London, 2007), pp. 255–268
18. B. Skoric, P. Tuyls, W. Ophey, in *Robust Key Extraction from Physical Unclonable Functions*. Applied Cryptography and Network Security (ACNS) 2005, Lecture Notes in Computer Science, vol. 3531 (Springer, Heidelberg, 2005), pp. 407–422
19. G.E. Su, S. Devadas, in *Physical Unclonable Functions for Device Authentication and Secret Key Generation*. Proceedings of the 44th Design Automation Conference, DAC 2007, San Diego, CA, USA, 4–8 June, (IEEE, Piscataway, NJ, 2007), pp. 9–14
20. P. Tuyls, G.J. Schrijen, F. Willems, T. Ignatenko, Secure key storage with PUFs, in *Security with Noisy Data - On Private Biometrics, Secure Key Storage and Anti-Counterfeiting* ed. by P. Tuyls, B. Skoric, T. Kevenaar. (Springer, London, 2007), pp. 269–292
21. P. Tuyls, B. Skoric, T. Kevenaar, *Security with Noisy Data. Private Biometrics, Secure Key Storage and Anti-Counterfeiting* (Springer, London, 2007)
22. P. Tuyls, G.-J. Schrijen, B. Skoric, J. van Geloven, N. Verhaegh, R. Wolters, in *Read-Proof Hardware from Protective Coatings*. Proceedings of the Eighth International Workshop on Cryptographic Hardware and Embedded Systems (CHES '06), Yokohama, Japan, 10–13 Oct, pp. 369–383
23. P. Tuyls, B. Skoric, S. Stallinga, A.H.M. Akkermans, W. Ophey, Information-theoretic security analysis of physical unclonable functions. in *Financial Cryptography and Data Security*, ed. by A.S. Patrick, M. Yung. 9th International Conference, FC 2005, Roseau, The Commonwealth of Dominica, 28 Febr –3 Mar 2005. Revised Papers, Lecture Notes in Computer Science, vol. 3570 (Springer, Berlin, Heidelberg, 2005), pp. 141–155

Towards Hardware-Intrinsic Security

Foundations and Practice

Sadeghi, A.-R.; Naccache, D. (Eds.)

2010, XVI, 407 p., Hardcover

ISBN: 978-3-642-14451-6