Security Analysis of Image-Based PUFs for Anti-counterfeiting

Saloomeh Shariati, François Koeune, François-Xavier Standaert

Security Analysis of Image-based PUFs for Anti-Counterfeiting

Saloomeh Shariati, François Koeune, and François-Xavier Standaert

ICTEAM Institute, Electrical Engineering Department
Université catholique de Louvain, Place du Levant, 3, B-1348
Louvain-la-Neuve,Belgium
saloomeh.shariati, francois.koeune, fstandae@uclouvain.be

Abstract. Physically Unclonable Functions are a promising tool to protect against counterfeiting attacks. Yet, as with any security system, it is important to embed them in a sound protocol, ensuring that no unexpected weakness is present in the "mortar" binding the components together. This paper proposes an anti-counterfeiting protocol that provably reduces to natural properties of its underlying components, namely an image-based Physical Function System bearing physical unclonability and an existentially unforgeable signature scheme. Experiments confirm the practical feasibility of our construction.

1 Introduction

Counterfeiting of trademarked products is a rapidly growing problem for the worldwide economy. Two types of threats are to be faced. On the one hand, Insider Counterfeiting (manufacturer overproduction) refers to unauthorized production by the legitimate manufacturer who realizes profits by producing extra quantities outside their license agreement [1]. On the other hand, Outsider Counterfeiting refers to unauthorized reproduction of products by other counterfeiters. Many ad-hoc methods have been proposed to avoid counterfeiting. Examples include so-called *overt* physical identifiers such as hologram and inks that visibly alter under light, or so-called *covert technology* such as invisible inks, proprietary photonic inks [2] and Radio Frequency Identification (RFID) Tags [3].

Recently [4, 5], Physically Unclonable Functions, or PUFs, have been introduced and, among many other applications, proposed as an effective tool for anti-counterfeiting systems. A Physically Unclonable Function (PUF) is a function that is embodied in a physical structure and is easy to evaluate, but hard to clone. Generally, a PUF interacts with stimuli (challenges) in an intricate way, and leads to unique and unpredictable responses. As a particular example, image-based PUFs are based on random visual features assessed by an imaging method. These can be employed for identification purposes [6].

For anti-counterfeiting applications, the core concept of using PUF primitives is to rely on the unique physical properties to identify a product. The PUF can be either intrinsic in the product or extrinsic and glued to the product. The general

idea is to digitally sign the product information (e.g. EPC code) together with the information extracted from the embedded PUF and use this signature as the certificate of authenticity [7–11].

Various image-based PUFs have been proposed to be applied in anti-counterfeiting systems, in different contexts and under different assumptions [10, 12–16]. In this paper, we propose a unified formal treatment of the use of image-based PUFs as a counterfeiting prevention tool. Starting with a description of an image-based Physical Function System, we define a secure anti-counterfeiting scheme and provide a construction meeting this definition. The construction combines physical protection blocks with cryptographic protection blocks. We define an attack model and derive the security property a PUF must fulfill in order to be eligible as a secure physical building block. We prove that this security property is equivalent to the physical unclonability property that was defined in [17]. We finally illustrate our model by studying a practical example.

The rest of the paper is structured as follows: Section 2 briefly describes previous works. In Section 3, we bring the previous approaches to the formalization of a Physical Function System and particularly image-based Physical Function System. In Section 4, we present an informal view of the anti-counterfeiting scheme and the security assumptions which is followed by the formal definition of anti-counterfeiting scheme in Section 5. Then we provide the attack model and formalization of the security of the anti-counterfeiting scheme in Section 6. Section 7 discusses application on a practical example.

2 Previous work

Early works that exploit the physical properties of random structures for authentication purposes date back to [18, 19]. The term Physically Unclonable Function was introduced by Pappu [4, 5]. Since then, many different physical objects have been proposed as PUF candidates, including Optical PUF [4, 5], Coating PUF [20], Silicon PUF [21–23], SRAM PUF [24], Paper PUF [10, 12–14], Phosphor PUF [3, 15], Laser-Written PUF [16, 25], etc.

Various application fields have also been proposed, such as secure key generation [26, 27], key storage [28], or in the design of block ciphers [29]. The idea of combining cryptographic means such as digital signature together with information extracted from the embedded PUF for authentication purpose was first applied in [7–11, 26].

3 Background

Armknecht et al. proposed a generic security framework of physical functions [17]. They explored the physical functions in general, where unclonability is only one possible security property. We briefly describe the components of the framework which will be used in anti-counterfeiting scheme afterwards. For a detailed description of each component we refer to [17]. A Physical Function (PF) consists of a *physical component* p and an *evaluation procedure* Eval. A

PF ($\mathsf{PF}_{p,\alpha_{\mathrm{PF}}} : \mathcal{X} \rightarrow \mathcal{Y}$) takes as input a challenge x and outputs a response y. The challenge-response behavior of a PF relies on the properties of the physical component p, an evaluation parameter α_{PF} and some evaluation noise (measurement uncertainties). A Physical Function is a probabilistic procedure because on a single challenge, it may produce different responses due to the evaluation noise. Since the output of the PF is noisy, usually it is combined with an *extraction algorithm* Extract with an extraction parameter α_{EX} that compensates a certain amount of noise and provides robust output. In addition to the response y, Extract also takes as input some *helper data* h generated the first time p was evaluated (i.e. in setup mode) and helping noise removal. The combination of PF and extraction algorithm is considered as one single building block which is defined as:

Definition 1 (Physical Function System [17]). *A* physical function system PFS *is a probabilistic procedure*

$$\mathsf{PFS}_{p,\alpha_{\mathrm{PF}},\alpha_{\mathrm{EX}}} : \mathcal{X} \times (\mathcal{H} \cup \{\epsilon\}) \rightarrow \mathcal{Z} \times \mathcal{H}, \tag{1}$$

where $\mathcal{X}$ is the set of challenges, $\mathcal{H}$ the set of helper data values, ϵ the empty string, and $\mathcal{Z}$ the set of outputs. Internally, a PF system is the combination of a physical function and an extraction algorithm Extract*, i.e.,*

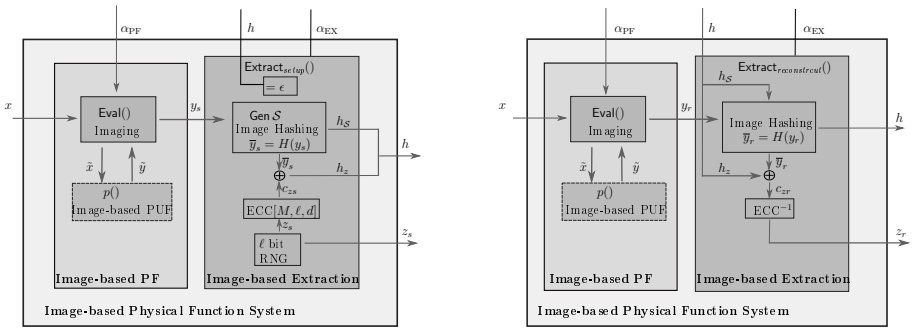
$$\begin{aligned} &\mathsf{PFS}_{p,\alpha_{\mathrm{PF}},\alpha_{\mathrm{EX}}}(x,h) \\ &= \mathsf{Extract}_{\alpha_{\mathrm{EX}}}(\mathsf{PF}_{p,\alpha_{\mathrm{PF}}}(x),h) \rightarrow (z,h') \end{aligned} \tag{2}$$

If $h \neq \epsilon$, then $h' = h$. Only in case $h = \epsilon$, a new helper data h' is generated for x. In the following, we omit the internal components and abbreviate $\mathsf{PFS} = \mathsf{PFS}_{p,\alpha_{\mathrm{PF}},\alpha_{\mathrm{EX}}}$.

Note that $h = \epsilon$ means that Extract should be executed in setup mode to generate a new helper data h w.r.t. challenge x. In case $h \neq \epsilon$, Extract should be executed in reconstruction mode to recreate output z associated with challenge x and helper data h.

In [6], the authors specialize the general Physical Function System for the specific case of image-based PUFs. The reason for specialization in image-based PUFs is twofold. First, the application of image-based PUFs is specific and is mainly for anti-counterfeiting systems. As a matter of fact, unlike some PUFs (e.g., optical PUFs), the input to image-based PUF is usually a fixed challenge and therefore a mathematical clone (a mathematical procedure that yields the same challenge-response behavior as the PUF e.g., a fake image) can be created by imitating the response of the PUF to this challenge (this is further discussed in Section 4.1). Secondly, the response of the image-based PUF is a real-valued image and require that a specific processing (i.e., dimensionality reduction and binarization) be integrated with the extraction procedure. Fig. 1 illustrates image-based PF system in setup and reconstruction mode.

An image-based PF System includes an image-based PF with a fixed challenge ($\mathcal{X} = x$) and an image-based Extraction. In set-up mode (Fig. 1(a)), image



(a) Setup mode. (b) Reconstruction mode.

Fig. 1. Image-based Physical Function system [6].

hashing compresses and binarizes the response of the image-based PF y_s and produces the image hash $\overline{y}_s$. Then a typical *fuzzy extraction* called Code-Offset fuzzy extraction [30] is applied. It masks the image hash $\overline{y}_s$ with a random code-word C_{zs} of a predefined Error Correcting Code (ECC) and generates output z_s and helper data h. In reconstruction mode, the response of the PF to the same challenge is evaluated y_r and image hashing generates image hash $\overline{y}_r$. Then ouput z_r is reconstructed using the image hash $\overline{y}_r$, second part of helper data h_z and ECC decoding as depicted in Fig. 1(b). For a more in depth view of the components of an image-based Physical Function System, we refer to [6].

4 Informal Description of Anti-Counterfeiting Scheme

In this section, a general view of the components of the anti-counterfeiting scheme and the security assumptions are described. The components of the scheme will be discussed in detail in the following section.

1. A configuration step Config is performed by the trademark owner and the PUF provider. They decide about the type of PUF, the parameters of the system, etc. A private/public key pair is also established in this step.
2. The PUF provider runs a creation process Create and delivers the created PUFs to the trademark owner.
3. A registration step Reg is performed by the trademark owner. It includes extracting digital information from the PUF and digitally signing this information plus some side information (e.g. serial number) about the product. Depending on the context, this step can take place either after the product to protect has been manufactured, or before: the PUF and registration data can for example be shipped to the manufacturer to be later physically or logically bound to the product. The PUF and data could for example be glued and printed on the product, or on a certificate of authenticity accompanying it. The product and PUF are then delivered to the market.

4. A verification process Verif is performed each time someone wants to determine whether the embedded PUF of a given product is authentic or not. Examples of entities performing the verification process includes: trademark owner, customs, wholesaler or retailer and end-user.

4.1 Security Assumptions

1. As a classical property of physically unclonable functions, we expect that, due to uncontrollable variations during the manufacturing process, PUFs cannot be physically cloned, i.e. that it is practically impossible/very costly for an adversary to generate pairs of PUFs yielding identical behavior when queried. We will provide an explicit phrasing of this expectation later in this paper (Eq. 8), and describe a testing methodology in Section 7.2. A PUF is usually employed to protect products whose value is less than the cost of cloning the PUF. As technology grows, we may expect that the cost of cloning a PUF does not anymore exceed the product value. As a consequence, the PUF-based system may need upgrade by time.
2. Image-based PUFs are physically unclonable and not necessarily mathematically unclonable. Mathematical unclonability means that it should be very hard to construct a mathematical procedure that yields the same challenge-response behavior as that of the PUF. Image-based PUFs in general do not have this property and a mathematical clone (e.g. a fake image) can be created by imitating the response (image) of the PUF. As a consequence, we assume that any verification process involves a prior verification that a real PUF, and not a mathematical clone (e.g. a picture) is being dealt with.
3. The various steps of the anti-counterfeiting scheme, i.e. Config, Create, Reg and Verif are performed by trusted parties, using trusted parameters. In particular, this implies that registration is only performed on physical components originating from a trusted source.
4. We authenticate the PUF and not the product. This is equivalent to authenticating the product itself when the PUF is inherently part of the product (*intrinsic* PUF). It is not necessarily the case when the PUF is a distinct object attached to the product (*extrinsic* PUF), but is still be sufficient in most anti-counterfeiting scenarios (both insider and outsider), where the trademark owner mostly wants to control the amount of products delivered to the market[1].

The above assumptions lay down a sound framework for implementing a secure PUF-based system. Let us briefly discuss some of their consequences.

As usual, we assume (assumption 1) that the PFS system in use generates inherently unique tags. However, it is difficult to ensure that, by modifying the generation parameters, an adversary will not be able to come up with a degraded

[1] For other scenarios, where an inseparable link between the PUF and the product is necessary, intrinsic PUFs, or additional measures such as using a tamper-proof seal would be necessary.

version of the PFS that will trigger collisions[2]. This could yield collision-based attacks, in a way very similar to hash-function collision attacks against signature schemes, where an adversary generates degraded tags, gets one of them registered, and can use this as a registration proof of the other ones. Checking whether a given tag was produced using the appropriate parameters is not always obvious for the registration authority or verifier. Assumption 3 allows us to get basically rid of that concern, as we assume that the only tags that will be registered are those produced by a trusted source, and thus implicitly using the correct parameters. We are thus left with the much more natural assumption that tags produced using different parameters would be sufficiently different to be sure that they cannot induce a collision with the "honest" ones. Of course, the validity of this "natural" assumption should still be asserted by the system designer when selecting a specific PUF realization.

It is worth noting that, although we assume above that the verification is performed by a trusted party, this trust is in fact only limited. An untrusted verifier is impossible to capture in a security model, in the sense that it is impossible to prevent a rogue verifier from simply providing a positive answer to any verification request. Nevertheless, the use of asymmetric cryptography allows storing only public, non-critical keys on the verifier's side. So a compromised verifier can (inevitably) be used to provide incorrect information to its user, but it cannot be exploited to affect the system's global security.

5 Formal Model of Anti-Counterfeiting Scheme

Let us now formally define our anti-counterfeiting scheme. In addition to the aforementioned PFS, it relies on a signature scheme providing existential unforgeability under an adaptive chosen messages attack. Intuitively, this means that it is impossible for an adversary running in "reasonable" time to produce the signature of a new message, even if he can beforehand request the signature of many messages that he chooses. We refer to [31] for a complete definition of this security property, which is a classical requirement of signature schemes.

Definition 2. *The anti-counterfeiting scheme Π includes tuple of four processes* $(\mathsf{Config}, \mathsf{Create}, \mathsf{Reg}, \mathsf{Verif})$ *satisfying the following:*

1. Config: The configuration process is performed one time and determines the type of PUF and the parameters of the system including the fixed challenge x, creation parameter α_{CR} , evaluation parameter α_{PF}, extraction parameter α_{EX} and a pair of private key sk and public key pk.
2. $\mathsf{Create}(\alpha_{\mathrm{CR}}) \rightarrow p$. The creation process takes as input the creation parameter α_{CR} and outputs the physical component p based on the creation parameter α_{CR}.

[2] As an extreme example, consider modifying an image-based PUF so that all produced tags are uniformly black.

3. $\mathsf{Reg}_{x,\alpha_{\mathrm{PF}},\alpha_{\mathrm{EX}},sk}(p, Aux) \rightarrow (\sigma, h, Aux)$. The registration process $\mathsf{Reg}_{x,\alpha_{\mathrm{PF}},\alpha_{\mathrm{EX}},sk}$ takes as input the physical component p and optional auxiliary data. Auxiliary data Aux includes side information about the PUF such as its serial number or EPC, creation date, expiration date, distribution points, *etc.* Since the challenge x, α_{PF} , α_{EX} and the private key sk are fixed and implicitly trusted, we usually discard them in our notation, that is we simply write Reg instead of $\mathsf{Reg}_{x,\alpha_{\mathrm{PF}},\alpha_{\mathrm{EX}},sk}$. The registration process Reg is performed in two phases:

   1) The set-up mode of the image-based PF system (See Fig. 1(a)) is executed for p using x, α_{PF} and α_{EX}, yielding the output z and helper data h as:

$$\mathsf{PFS}(x, \epsilon) \rightarrow (z, h) \tag{3}$$

   2) The output, helper data and auxiliary data are signed using the private key sk, yielding the signature $\sigma = \mathsf{Sign}_{sk}(z||h||Aux)$.

4. $\mathsf{Verif}_{x,\alpha_{\mathrm{PF}},\alpha_{\mathrm{EX}},pk}\ (p, h, Aux, \sigma) \rightarrow b \in \{0, 1\}$. The verification process $\mathsf{Verif}_{x,\alpha_{\mathrm{PF}},\alpha_{\mathrm{EX}},pk}$ takes as input a physical component p, helper data h, auxiliary data Aux and signature σ, and outputs a validity bit b. Hereafter, we simply write Verif instead of $\mathsf{Verif}_{x,\alpha_{\mathrm{PF}},\alpha_{\mathrm{EX}},pk}$. The verification process Verif is also performed in two phases:
   1) The reconstruction mode of image-based PF system (See Fig. 1(b)) is executed for p using x, α_{PF}, α_{EX} and the helper data h. The output z_r is generated as following:

$$\mathsf{PFS}(x, h) \rightarrow (z_r, h) \tag{4}$$

   2) The verification algorithm of the signature scheme is executed on $(z_r||h||Aux)$ using the public key pk to check the authenticity of the PUF. The verification algorithm of the signature scheme outputs a bit b, with $b = 1$ meaning signature valid and $b = 0$ meaning signature invalid.
   The verification process outputs the bit b.

6 Attack Model and Security

For our security analysis, we assume that the Adversary A has access to an oracle $\mathsf{CReg}(.)$. When it is queried with parameter $\alpha_{\mathrm{CR,i}}$, this oracle checks whether $\alpha_{\mathrm{CR,i}} \in \mathcal{A}_{\mathrm{CR}}$ and, if it does, creates p_i, registers the created p_i and returns $(p_i, \sigma_i, h_i, Aux_i)$ (Fig. 2).

In practice, only adversaries with bounded time and computational effort are relevant. Thus we consider only PPT adversaries and additionally limit them to query the CReg oracle at most q times.

Definition 3 (Attack Model). *Let* $\Pi = (\mathsf{Config}, \mathsf{Create}, \mathsf{Reg}, \mathsf{Verif})$ *be the anti-counterfeiting scheme. Adversary* A *is given pk and oracle access to* $\mathsf{CReg}(.)$ *as defined above. Let* Q *be the set of pairs* (p, Aux) *that were generated by*

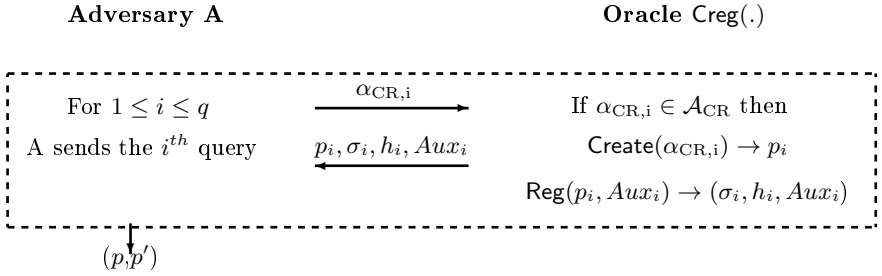


Fig. 2. Counterfeiting experiment $\mathsf{Counterfeit}_{\mathsf{A},\Pi}(q)$.

$\mathsf{CReg}(.)$. *The adversary* A *then generates a set of* (p', h', Aux', σ'). *The output of the counterfeiting experiment* $\mathsf{Counterfeit}_{\mathsf{A},\Pi}(q)$ *is defined to be* 1 *if*

$$\mathsf{Verif}(p', h', Aux', \sigma') \rightarrow 1 \quad \text{and} \quad (p', Aux') \notin Q. \tag{5}$$

The security of the anti-counterfeiting scheme is then defined as follows:

Definition 4 (Security). *The anti-counterfeiting scheme* Π *is* β*-secure if for all probabilistic polynomial-time (PPT) adversaries, we have:*

$$Pr\Big[\mathsf{Counterfeit}_{\mathsf{A},\Pi}(q) = 1\Big] \leq \beta \tag{6}$$

Lemma 1. *Suppose that the signature scheme is existentially unforgeable under an adaptive chosen message attack. If the output of the adversary experiment is* 1*, then there exists at least one instance* p_i *such that there has been a registration step:*

$$\mathsf{Reg}(p_i, Aux_i) \rightarrow (\sigma_i, h_i, Aux_i) \tag{7}$$

and $\sigma' = \sigma_i$, $h' = h_i$, $Aux' = Aux_i$ *and* $z' = z_i$ *s.t* $\mathsf{PFS}(x, h_i) \rightarrow (z', h_i)$.

Proof. Suppose that there does not exist any p_i *that has been registered by a trusted authority such that* $\sigma' = \sigma_i$, $h' = h_i$, $Aux' = Aux_i$ *and* $z' = z_i$. *Then,* $\mathsf{Verif}(p', h', Aux', \sigma', pk) \rightarrow 1$ *implies that the adversary succeed in producing a valid signature pair* $((z'||h'||Aux'), \sigma')$ *for a signature that has not been produced by the signature scheme. This contradicts the existential unforgeability assumption for the signature scheme.*

As a direct consequence of Lemma 1, the only way for the adversary to make a counterfeit is to produce a physical component p' that *collides* with a previously registered physical component p_i, i.e. provides the same output using its corresponding helper data h_i. Hence, the security property of the anti-counterfeiting scheme can be evaluated as follows:

Property 1. The anti-counterfeiting scheme Π is β-secure (w.r.t. x) if for all PPT adversaries A that are limited to calling the CReg oracle at most q times, it holds that:

$$Pr\Big[\mathsf{PFS}'(x,h) \to (z,h) : \mathsf{PFS}(x,\epsilon) \to (z,h) \\ \mathsf{Counterfeit}_{\mathsf{A},\Pi}(q) \to (p,p'); \\ p \in [\mathsf{Create}(\alpha_{\mathrm{CR}} \in \mathcal{A}_{\mathrm{CR}})]; p' \in [\mathsf{Create}(\alpha'_{\mathrm{CR}} \in \mathcal{A}_{\mathrm{CR}})]\Big] \leq \beta \tag{8}$$

where ':' denotes the conditional probability and PFS' is the PF system using p'. The security experiment $\mathsf{Counterfeit}_{\mathsf{A},\Pi}(q)$ is depicted in Fig. 2.

This corresponds to the existential physical unclonability property that was defined in [17].

7 Implementation

Let us illustrate this on an example. For this purpose, we first need to briefly discuss another property, namely, the construction's *robustness*.

7.1 Robustness

So far, we emphasized on the security of the anti-counterfeiting system. However, for the system to be useful in practice, we should also ensure that outputs are reproducible, i.e. that different evaluations of a single PUF produce the same output. Robustness [17] expresses the probability that the output generated by the reconstruction phase matches the value generated in the set-up phase and is formally defined as:

Definition 5 (Robustness). *Let* PFS *be a PF system (Definition 1) and let* $x \in \mathcal{X}$ *be a challenge. The* robustness *of* PFS *(w.r.t. x) is defined as the probability*

$$\rho_{\mathsf{PFS}}(x) := \Pr\left[\mathsf{PFS}(x,h) \to (z,h) : \mathsf{PFS}(x,\epsilon) \to (z,h)\right] \tag{9}$$

A sound evaluation of a practical anti-counterfeiting system requires assessing concurrently its security and its robustness.

7.2 Testing Method

As discussed in Section 4, the assumption that only tags produced by a trusted manufacturer are registered can often simplify unclonability testing. Relying on

the assumption that different creation parameters will not trigger collisions with tags produced using α_{CR}, we can rewrite Eq. 8 as:

$$Pr\Big[\mathsf{PFS}'(x,h) \to (z,h) : \mathsf{PFS}(x,\epsilon) \to (z,h) \\ (p,p') \in [\mathsf{Create}(\alpha_{\mathrm{CR}} \in \mathcal{A}_{\mathrm{CR}})] \leq \beta; \tag{10}$$

and thus focus on the probability that an honest manufacturer creates two clones by coincidence.

Following the method of [6], this probability can be statistically estimated by sampling R PUFs and their corresponding pictures. First, one PUF is chosen as a "target", the image of which is used in set-up mode to generate helper data h and output z_s. The remaining $(R-1)$ images, together with the initial helper data h, are then used in reconstruction mode, and the generated outputs are compared with z_s. This experiment is repeated R times, each PUF being selected once as target. The security property β is then estimated by:

$$Pr\Big[z_r = z_s : p \neq p'\Big] \leq \beta \tag{11}$$

Similarly, robustness can be evaluated from Eq. 9, using a dataset of PUFs that are evaluated several times with different observation noise. Given a dataset of P different PUFs observed Q times each, the robustness can be statistically estimated as follows. For each PUF, one observation of the PUF is used in set-up mode to produce output z_s and helper h. The remaining $(Q-1)$ observations, together with the initial helper data h, are then used in reconstruction mode, and the generated outputs are compared with z_s. This is repeated Q times, with each observation being once in the set-up phase. The robustness is then estimated from $\Pr[z_r = z_s]$ that is equivalent to Eq. 9.

7.3 Practical Example

As a practical example, we implemented an image-based Physical Function System based on a Laser-Written PUF (LPUF). The basic principle consists in engraving tiny laser marks on the surface or volume of a transparent object. Due to instabilities in the laser beam and small variations in the matter of the object, the engraved mark will bear random characteristics that are very difficult to reproduce[3]. LPUF is a good instance of image-based PUFs to be employed for anti-counterfeiting purposes. Indeed, it can be engraved in several objects with various materials and it can be very small. It is also robust against aging especially when embedded on the bulk of the object.

For evaluation purpose, we manufactured samples with engraved marks of diameter 60μm. Assuming a locality principle, i.e. that random variations will behave as independent events when occurring at different locations on the object, security can be improved by increasing the size of the mark or by engraving multiple marks on a physical object.

[3] We refer to [6] for a detailed description of the process.

In our evaluation, we set extraction parameters providing a robustness level of 94% for a dataset of 20 different LPUFs observed 100 times each[4]. The security property β is then evaluated using another dataset containing $R = 1000$ different LPUFs observed one time. Using the approach described in Section 7.2, the probability of collisions between outputs of different PUFs (Eq.11) is estimated to be $\beta = 10^{-5}$. The birthday paradox theory states that if q elements are drawn from a discrete uniform distribution with range $[1, d]$, the probability of collision is $1 - \left(\frac{d-1}{d}\right)^{q.(q-1)/2}$. Applying this formula with $d = 1/\beta$ allows computing the probability of collision for a given number q of PUFs.

Based on the aforementioned locality principle, this means we could obtain a reasonable security level by increasing the size of the mark (or the number of marks) by a factor between 4 and 16. Considering the 60μm diameter of the original mark, this seems a very practical option.

We also compare the obtained results with the typical results given by human biometrics. In biometrics, e.g., for fingerprints verification systems, the performance of the system is often evaluated by False Rejection Rate (FRR), False Acceptance Rate (FAR) and Equal Error Rate (EER). FRR is the probability for a valid user to be incorrectly rejected and FAR is the probability of an imposter to be incorrectly matched to the biometric of a valid user. EER is the rate at which both FAR and FRR are equal. Therefore, for the image-based physical function system, FRR and FAR can be considered to be equivalent to $1 - \rho$ and β respectively. The equivalent ERR is given when $1 - \rho = \beta$ and is obtained as $EER_g = 0.2\%$. In this view, our example of image-based physical function system provides better results with respect to typical fingerprints verification systems ($EER > 2\%$).

8 Conclusion

The protocol we propose is simple and reasonably efficient. It relies on a natural property that can be expected from any image-based Physical Function System, namely the fact that each object presents a unique, typical visual aspect. This is presumably easier to achieve in practice than more advanced properties such as the availability of a large amount of independent challenge-response pairs. From a computational point of view, product registration (resp. verification) requires one execution of an asymmetric cryptographic primitive, which is in line with the cost and computing capabilities expected from an image processing-capable device.

The protocol allows a simple key management policy, where critical, private keys can be kept under control of the trademark owner, whereas only public keys need to be distributed towards the (more difficult to control) verifiers. It also

[4] For the sake of completeness, we note that the extraction parameters are set as follows: Gabor hashing parameters are fixed to (ν_0, F, a, Δ)=(π/3,4,10,30) and a BCH (255,131,18) is used for fuzzy extraction. This yields the binary output z with 131 bits. For further details on the definitions of parameters, we refer to [6].

allows straightforward extensions, e.g. integrating a Public Key Infrastructure (PKI) for better scalability.

Experiments confirm that some Physical Function Systems, such as laser engraving, seem to bear the necessary properties to be integrated in our construction and provide a practical anti-counterfeiting system.

Eventually, our results also confirm that the security framework in [17] can be used to bridge the gap between engineering constraints and cryptographic protocols. While the previous work in [6] shows that metrics such as unclonabilibilty and robustness can indeed be estimated for real-world PUFs, the security analysis in this paper confirms that these metrics can also be connected to standard cryptographic analyses, using sound (and tight) reductions.

Acknowledgments

This research work was supported by the Belgian Walloon Region project TRACEA. François-Xavier Standaert is an Associate Researcher of the Belgian Fund for Scientific Research (FNRS-F.R.S.). This work has been funded in part by the ERC project 280141 (acronym CRASH).

References

1. T. Staake and E. Fleisch. *Countering Counterfeit Trade: Illicit Market Insights, Best-Practice Strategies, and Management Toolbox*. Springer Publishing Company, Incorporated, 1st edition, 2010.
2. S. Bastia. Next generation technologies to combat counterfeiting of electronic components. *IEEE Trans. on Components and Packaging Tech.*, 25:175–176, 2002.
3. C. N. Chong et al. Anti-counterfeiting with a random pattern. In *Int. Conf. on Emerging Security Information, Systems and Tech.*, pages 146–153, 2008.
4. R. Pappu. *Physical one-way functions*. PhD thesis, MIT, March 2001.
5. R. Pappu et al. Physical one-way functions. *Science*, 297, 2002.
6. S. Shariati, F-X. Standaert, L. Jacques, and B. Macq. Comprehensive study of image-based physical function system. *Submitted to Journal of Cryptographic Engineering*.
7. P. Tuyls and B. Škorić. Strong authentication with physical unclonable functions. In *Security, Privacy, and Trust in Modern Data Management*, pages 133–148. 2007.
8. P. Tuyls et al. Secure key storage and anti-counterfeiting. *Springer*, pages 255–268, 2008.
9. P. Tuyls and L. Batina. Rfid-tags for anti-counterfeiting. In *Topics in Cryptology - CT-RSA 2006, volume 3860 of LNCS*, pages 115–131. Springer Verlag, 2006.
10. P. Bulens, F.-X. Standaert, and J.-J. Quisquater. How to Strongly Link Data and its Medium: the Paper Case. *IET Information Security*, 4(2):125–136, 2010.
11. D. Kirovski. Anti-counterfeiting: Mixing the physical and the digital world. In Jorge Guajardo, Bart Preneel, Ahmad-Reza Sadeghi, and Pim Tuyls, editors, *Foundations for Forgery-Resilient Cryptographic Hardware*, number 09282 in Dagstuhl Seminar Proceedings, 2010.
12. J. D. R. Buchanan, R. P. Cowburn, A. V. Jausovec, D. Petit, P. Seem, G. Xiong, D. Atkinson, K. Fenton, D. A. Allwood, and M. T. Bryan. Fingerprinting documents and packaging. *Nature*, page 475, 2005.

13. D. Kirovski. Toward an automated verification of certificates of authenticity. In *Proceedings of the 5th ACM conference on Electronic commerce*, EC '04, pages 160–169. ACM, 2004.
14. Y. Chen, K. Mihçak, and D. Kirovski. Certifying authenticity via fiber-infused paper. *SIGecom Exch.*, 5:29–37, April 2005.
15. C. N. Chong and D. Jiang. Anti-counterfeiting using phosphor puf. In *International Conference on In Anti-counterfeiting*, pages 59–62, 2008.
16. S. Shariati, F-X. Standaert, L. Jacques, B. Macq, M. A. Salhi, and P. Antoine. Random profiles of laser marks. In *WIC Symposium on Information Theory in the Benelux*, pages 27–34, 2010.
17. F. Armknecht, R. Maes, A. R. Sadeghi, F. X. Standaert, and C. Wachsmann. A formalization of the security features of physical functions. In *IEEE Symposium on Security and Privacy*, pages 397–412, 2011.
18. D.W. Bauder. An anti-counterfeiting concept for currency systems. Technical Report PTK-11990, Sandia National Labs, Albuquerque, NM, 1983.
19. Commission on Engineering Committee on Next-Generation Currency Design and National Research Council Technical Systems. *Counterfeit Deterrent Features for the Next-Generation Currency Design*. The National Academies Press, 1993.
20. P. Tuyls, G. Jan Schrijen, B. Skoric, J. van Geloven, N. Verhaegh, and R. Wolters. Read-proof hardware from protective coatings. In *Cryprographic Hardware and Embedded Systems Workshop*, volume 4249 of *LNCS*, pages 369–383. Springer, 2006.
21. B. Gassend, D. Clarke, M. van Dijk, and S. Devadas. Silicon physical random functions. In *ACM conference on Computer and communications security*, November 2002.
22. 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*, 13(10):1200–1205, 2005.
23. J. W. Lee, Daihyun Lim, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas. A technique to build a secret key in integrated circuits for identification and authentication applications. In *VLSI Circuits. Digest of Technical Papers*, pages 176–179, 2004.
24. J. Guajardo, S. S. Kumar, G. Jan Schrijen, and P. Tuyls. FPGA intrinsic PUFs and their use for IP protection. In *Cryprographic Hardware and Embedded Systems Workshop*, volume 4247, pages 63–80, 2007.
25. S. Shariati, L. Jacques, F-X. Standaert, B. Macq, M. A. Salhi, and P. Antoine. Randomly driven fuzzy key extraction of uncloneable images. In *International Conference on Image Processing (ICIP)*, 2010.
26. P. Tuyls and B. Skoric. Secret key generation from classical physics. In *Philips Research Book Series*, 2005.
27. B. Skoric, P. Tuyls, and W. Ophey. Robust key extraction from physical uncloneable functions. In *Applied Cryptography and Network Security (ACNS)*, pages 407–422, 2005.
28. D. Lim, J. W. Lee, B. Gassend, G. Edward Suh, M. van Dijk, and S. Devadas. Extracting secret keys from integrated circuits. *IEEE Transactions on VLSI Systems*, 13(10):1200–1205, 2005.
29. F. Armknecht, R. Maes, A. R. Sadeghi, B. Sunar, and P. Tuyls. Memory leakage-resilient encryption based on physically unclonable functions. In *Advances in Cryptology (ASIACRYPT)*, volume 5912 of *LNCS*, pages 685–702, 2009.
30. Y. Dodis et al. Fuzzy extractors: How to generate strong secret keys from biometrics and other noisy data. In *Eurocrypt'04*, pages 523–540, 2004.

31. J. Katz and Y. Lindell. *Introduction to Modern Cryptography.* Chapman and Hall/CRC Press, 2008.