Robust key extraction from Physical Uncloneable Functions

B. Škorić, P. Tuyls, W. Ophey

Philips Research Laboratories
Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands

Physical Uncloneable Functions (PUFs) can be used as a cost-effective means to store key material in an uncloneable way. Due to the fact that the key material is obtained by performing measurements on a physical system, noise is inevitably present in each readout. In this paper we present a number of methods that improve the robustness of bit-string extraction from noisy PUF measurements in general, and in particular for optical PUFs. We describe a practical implementation in the case of optical PUFs and show experimental results.

Keywords: Physical Uncloneable Function, authentication, speckle pattern, Challenge-Response Pair, noise, error correction

1 Introduction

1.1 General introduction to PUFs

A 'Physical Uncloneable Function' (PUF) is a function that is realized by a physical system, such that the function is easy to evaluate but the physical system is hard to characterize, model or reproduce.

Physical tokens were first used as identifiers in the 1980s in the context of strategic arms limitation treaty monitoring [1]. The concept was investigated for civilian purposes in the 1990s [2]. The tokens which were then studied are very hard to reproduce physically, but quite easy to read out completely, i.e. all the physical parameters necessary for successful identification are readily given up by the token. This makes these tokens suitable for systems where the verifier knows with certainty that an actual token is being probed and that the measuring device can be trusted. However, the tokens are not suitable for online identification protocols with an invisible party. An imposter can easily copy the data from someone's token, and then enter that data through a keyboard. The verifier cannot see the difference between the real token and the cloned data.

Truly uncloneable tokens (PUFs) were introduced by Pappu [3, 4]. These are so complex that it is infeasible to fully read out the data contained in a token or to make a computer model that predicts the outputs of a token [5]. This makes PUFs suitable for online protocols as well as verification involving physical probing by untrusted devices.

A PUF is a physical system designed such that it interacts in a complicated way with stimuli (*challenges*) and leads to unique but unpredictable *responses*. A PUF challenge and the corresponding response are together called a *Challenge-Reponse-Pair* (CRP). A PUF behaves like a keyed hash function; The physical

system consisting of many 'random' components is equivalent to the key. In order to be hard to characterize, the system should not allow efficient extraction of the relevant properties of its interacting components by measurements. Physical systems that are produced by an uncontrolled production process, e.g. random mixing of several substances, turn out to be good candidates for PUFs. Because of this randomness, it is hard to produce a physical copy of the PUF. Furthermore, if the physical function is based on many complex interactions, then mathematical modeling is also very hard. These two properties together are referred to as *Uncloneability*.

1.2 Applications

From a security perspective the uniqueness of the responses and uncloneability of the PUF are very useful properties. Because of these properties, PUFs can be used as unique identifiers, means of tamper-detection and/or as a cost-effective source for key generation (common randomness) between two parties. By embedding a PUF inseparably into a device, the device becomes uniquely identifiable and uncloneable. Here 'inseparable' means that any attempt to remove the PUF will with very high probability damage the PUF and destroy the key material it contains. A wide range of devices can be equipped with a PUF in this way, e.g. smart-cards, credit cards, RFID tags, value papers, optical discs (DRM), chips, security cameras, etc.

An identification scheme based on CRPs works as follows. First, one needs a detector for measuring the analog output of a PUF and an algorithm that extracts bit-strings from this output. The detector and the processor executing the algorithm can be located on the device with the embedded PUF, or inside a separate external reader device. The scheme consists of two phases: *enrollment* and *verification*. In the enrollment phase, the Verifier produces the PUF, embeds it in a device, and stores an initial set of CRPs securely in his database. Then the device is given to a user. The verification phase starts when the user presents his device to a terminal. The Verifier sends a randomly chosen PUF challenge from his database to the user. If the Verifier receives the correct answer[1] from the device, the device is identified. Furthermore, a secure authenticated channel can be set up between the verifier and the device, using a session key based on the PUF response.

A special class of applications becomes possible if so-called 'control' is introduced [6]. A *Controlled PUF* (CPUF) is a PUF that is bound to a processor which completely governs the input and output. The chip can prohibit frequent challenging of the PUF and forbid certain classes of challenge. It can scramble incoming challenges. Furthermore, it can hide the physical output of the PUF, revealing to the outside world only indirect information derived from the output, e.g. an encryption or hash. This control layer substantially strengthens the security, since an attacker cannot probe the PUF at will and cannot interpret

[1] In general, the 'answer' is the result of cryptographic operations involving the PUF response. For details on secure protocols we refer to [6, 7, 9].

the responses. CPUFs allow for new applications such as 'certified execution' [6, 7] and 'certified measurement'.

1.3 Types of PUF / Physical realizations

Several physical systems are known on which PUFs can be based. The main types are optical PUFs [3, 4], coating PUFs [7], silicon PUFs [8, 9] and acoustic PUFs [7]. In this paper we first discuss PUFs in general and then focus on optical PUFs.

Optical PUFs consist of a transparent material containing randomly distributed scattering particles. Their suitability as a carrier of secret key material derives from the uniqueness and unpredictability of speckle patterns that result from multiple scattering of laser light in a disordered optical medium [5]. The challenge can be e.g. the angle of incidence, focal distance or wavelength of the laser beam, a mask pattern blocking part of the laser light, or any other change in the wave front. The output is the speckle pattern. As the speckle pattern contains many randomly distributed bright and dark patches, a high-entropy bit-string can be extracted from it, using a modest amount of image analysis. Physical copying of optical PUFs is difficult for two reasons: (i) The light diffusion obscures the locations of the scatterers. At this moment the best physical techniques can probe diffusive materials up to a depth of approximately 10 scattering lengths [10]. (ii) Even if all scatterer locations are known, precise positioning of a large number of scatterers is very hard and expensive, and requires a production process different from the original randomized process. *Modeling*, on the other hand, is difficult due to the inherent complexity of multiple coherent scattering [11]. Even the 'forward' problem turns out to be hard. Given the details of all the scatterers, the fastest known computation method of a speckle pattern is the transfer-matrix method [12]. It requires in the order of $(A/\lambda^2)^3 d/\lambda$ operations (where A is the illuminated area, λ the wavelength and d the PUF thickness), which is larger than 10^{20} even if rather conservative values are chosen for A, λ and d.

1.4 The robustness problem

The main problem facing any non-digital data storage mechanism is reproducibility. Due to the inherent noisiness of physical measurements, a readout will never yield exactly the same result.

1. For uncontrolled PUFs the external reader that challenges the PUF and detects the response during the verification phase can be a different device than the one that was used in the enrollment phase. Alignment and sensitivity differences between readers give rise to noise, unless great pains are taken to enforce very small mechanical and/or electrical tolerances. However, the potential number of readers is enormous, making such a standardisation impractical and expensive. Hence, the inter-device deviations give an important contribution to the noise in the readout of uncontrolled PUFs.

2. Even repeated measurements with the same challenging and detection device do not give identical results. Time dependent external influences like temperature, moisture, vibrations, stray light, stray fields etc. can have an impact on the measurements.
3. The PUF itself is not immutable. It can accidentally get damaged. Another problem is spontaneous degradation. Most materials slowly change over time due to chemical reactions, friction and repeated thermal deformations. The rate of drifting determines the lifetime of the key material in the PUF.

Robustness can be achieved in two ways, which are best combined: (a) Reducing the noise at the source, and (b) Given a certain level of noise, extracting as much robust key material as possible by properly choosing an error correction algorithm. In Section 2 general measures are discussed to achieve both these goals. They apply to all types of PUF. The methods in Sections 2.3 and 2.4 are new. In Section 3 we present noise reduction methods for optical PUFs. In Section 4 we show experimental results on key extraction in the case of optical PUFs.

2 Key Extraction from Noisy Data

2.1 Shielding functions

Generally speaking a key extraction algorithm is built on a Secret Extraction Code [13] or, equivalently, a Fuzzy Extractor [2] [14]. For the sake of simplicity we describe the algorithm in terms of a *shielding function* [16], which generates a special set of Secret Extraction Codes, while having all the necessary properties. We denote the analog PUF response to a challenge C during the enrollment phase by $R \in \mathbb{R}^n$ and during the verification phase by $R' \in \mathbb{R}^n$. A function $G : \mathbb{R}^n \times \mathcal{W} \to \{0,1\}^k$ is called δ-contracting if for all R there exists at least one element $W_C \in \mathcal{W}$ and $K \in \{0,1\}^k$ such that $G(R', W_C) = G(R, W_C) = K$ for all R' that lie within a sphere with radius δ around R (i.e. $||R' - R|| \leq \delta$). We use δ-contracting functions to extract keys $K = G(R, W_C)$ from noisy data R using *helper data* W_C.

The function $G(\cdot,\cdot)$ is called 'versatile' if the sets $S_G(R) = \{K \in \{0,1\}^k \mid \exists W_C$ such that $G(R, W_C) = K\}$ are sufficiently large for sufficiently many R.

A function $G : \mathbb{R}^n \times \mathcal{W} \to \{0,1\}^k$ is called ε-revealing if W_C leaks less than ε bits on K (in the information theoretic sense), i.e. $\mathbf{I}(W_C; K) \leq \varepsilon$. An (ε, δ)-shielding function $G : \mathbb{R}^n \times \mathcal{W} \to \{0,1\}^k$ is a function that is δ-contracting, versatile and ε-revealing. It is used to extract a secret of length k from the PUF response as follows.

- **Enrollment Phase:** The PUF is subjected to a challenge C and the analog response R is measured. Then a random key K is chosen from $\{0,1\}^k$ and helper data W_C is computed by solving $G(R, W_C) = K$ for W_C. The quadruplet $(\mathrm{ID}_{\mathrm{PUF}}, C, W_C, K)$ is then stored in a database.

[2] A special case of this construction was previously considered in [15] in the context of biometrics, where it was called a 'fuzzy commitment'.

– **Verification Phase:** When the PUF is inserted into the reader the PUF's identity is sent to the Verifier. The Verifier chooses a random challenge C from his database and sends it to the PUF together with the corresponding helper data W_C. Then the reader subjects the PUF to the challenge C and measures its response R'. The reader computes a key $K' = G(R', W_C)$.

It follows from the δ-contracting property of the function G that $K' = K$ if R' is sufficiently close to R.

In the case of analog outputs, $G(\cdot, \cdot)$ will typically comprise a quantisation procedure. If the strings obtained after quantisation are uniformly distributed, the distilled keys K can be used securely (the helper data leaks no information on K). However, if those strings are not uniformly distributed, a privacy amplification like step, e.g. based on universal hash functions, has to be applied to obtain a (shorter) key about which the adversary has only a negligible amount of information.

2.2 Example algorithm

In order to illustrate the above definitions we present an example based on an Error Correcting Code $\mathcal{E}$. The algorithm makes use of so-called 'robust components', which are parts of the PUF response that are observed to be relatively insensitive to noise during enrollment. These are e.g. parts of the analog response R whose magnitude exceeds a certain threshold, or parts that do not strongly vary when the measurement is repeated a number of times. By A/D converting R, a 'raw' bit-string $\boldsymbol{b}$ is obtained. Substrings in $\boldsymbol{b}$ that correspond to robust components in R are referred to as 'robust bits'.

– **Enrollment Phase:** The PUF is subjected to a challenge C. The analog output is converted into a bit-string $\boldsymbol{b}$. Robust components are determined, and a set $\mathcal{I}$ is constructed, consisting of indices pointing at the locations of the robust bits in $\boldsymbol{b}$. The so-called *robust bit string* X is obtained by concatenating the robust bits. Then a *secret* key K is randomly generated and encoded to a code word $S_K \in \mathcal{E}$. The difference $W = X \oplus S_K$ is computed. The total set of helper data consists of the set $\mathcal{I}$ and the string W. The Verifier stores $(\mathrm{ID}_{\mathrm{PUF}}, C, \mathcal{I}, W, K)$.
– **Verification Phase:** When the PUF is inserted into the reader the PUF's identity is sent to the Verifier. The Verifier chooses a random challenge C from his database and sends it to the reader together with the corresponding helper data $\mathcal{I}, W$. The reader subjects the PUF to the challenge C and converts the analog response R' into a bit-string $\boldsymbol{b}'$. It uses the helper data indices $\mathcal{I}$ to select bits from $\boldsymbol{b}'$, yielding a bit-string X'. It uses the second part of the helper data, W, to compute $S' = X' \oplus W = (X' \oplus X) \oplus S_K$. Finally, it employs $\mathcal{E}$ to correct any errors present in S'.

Clearly, if the number of errors is not too large ($X' \approx X$) then the error-correcting code will properly correct S' into S_K and yield K after decoding. Note that the δ-contracting property arises from the error correcting capacity of

$\mathcal{E}$, while the ε-revealing property follows from the fact that the secret S_K gets masked by the random variable X.

2.3 Calibration CRPs

In uncontrolled PUFs, the main source of noise is misalignment of the challenging apparatus. We describe a method to reduce this misalignment. A small number of CRPs is reserved for calibration purposes, and is never used for identification. The protocol works as follows.

- **Enrollment of Calibration CRPs:** In addition to the 'ordinary' enrollment, a number of *Calibration CRPs* $(C_{\text{cal}}, r_{\text{cal}})$ is measured and stored. (Here the notation r_{cal} stands for information about the response in general; r_{cal} does not have to be of the same type as the 'ordinary' response information that is stored for identification purposes). The Calibration CRPs have no challenges in common with the 'ordinary' CRPs. The Calibration CRPs are not secret and hence they can be stored in a publicly accessible way, e.g. next to the PUF.
- **Use of Calibration CRPs in the Verification Phase:** A PUF is inserted into a reader. The reader reads ID_{PUF} and acquires a Calibration CRP $(C_{\text{cal}}, r_{\text{cal}})$ corresponding to ID_{PUF}. (This CRP is obtained e.g. by reading it from the smart-card which contains the PUF, or the CRP is sent by the Verifier). The PUF is subjected to the challenge C_{cal}, and the response r'_{cal} is measured. Based on the difference between r'_{cal} and r_{cal}, the alignments of the reader are adjusted. The process of measuring the response to C_{cal} and adjustment is repeated until the difference between r'_{cal} and r_{cal} is reduced to an acceptable level. Only if this level is reached, the Verifier sends a challenge C intended for identification purposes, and the 'real' identification protocol as described in Section 1.2 starts running.

There are ways to improve this method. One option is to choose the calibration challenges such that identification challenges are never extremely far away from a calibration point. In this way the error introduced by moving away from a calibration point is reduced. Another option is to subdivide the process of looking for the correct settings into several stages: First a coarse search with low discriminating power, and then a finer search. In optical PUFs, the discriminating power can be adjusted by changing the laser beam diameter. The sensitivity to noise decreases with increasing beam diameter.

The search can be accelerated by storing additional 'perturbed' responses during enrollment. Pairs $\{\boldsymbol{\Delta}_i, R_{\text{cal}}(\boldsymbol{m} - \boldsymbol{\Delta}_i)\}$ are stored together with the CRP $(C_{\text{cal}}, R_{\text{cal}}(\boldsymbol{m}))$, where $\boldsymbol{m}$ denotes the correct settings of the reader, and $\boldsymbol{\Delta}$ a small perturbation. When, during the search, a response matches $R_{\text{cal}}(\boldsymbol{m} - \boldsymbol{\Delta}_i)$, the reader knows that its settings must be adjusted by an amount $\boldsymbol{\Delta}_i$.

2.4 Two-way use of helper data

In all schemes discussed so far, helper data is generated during enrollment and applied at the time of verification. However, the measuring device is capable of

producing helper data also in the verification phase. Instead of discarding this extra information, one can use it to improve the robustness of the extracted keys. We present an interactive protocol in which the robust components obtained from enrollment and verification are combined using an 'AND' operation.

- **Enrollment:** The Verifier subjects the PUF to a challenge C and converts the analog response R to a bit-string $\boldsymbol{b}$. He determines robust components and constructs the helper data set $\mathcal{I}$ of pointers to the robust parts of $\boldsymbol{b}$. He stores $(\mathrm{ID}_{\mathrm{PUF}}, C, \mathcal{I}, \boldsymbol{b})$.
- **Verification:** The PUF is inserted into the reader and the reader sends $\mathrm{ID}_{\mathrm{PUF}}$ to the Verifier. The Verifier sends C and $\mathcal{I}$. The reader challenges the PUF with C and measures a response R', which it converts into a bit-string $\boldsymbol{b}'$. It determines the robust components of R' and constructs new helper data $\mathcal{I}'$. It sends $\mathcal{I}'$ to the Verifier. Both the reader and the Verifier now compute the *combined helper data* $\mathcal{J} = \mathcal{I} \cap \mathcal{I}'$. The Verifier computes $X = \boldsymbol{b}_{\mathcal{J}}$, while the reader computes $X' = \boldsymbol{b}'_{\mathcal{J}}$. (The notation $\boldsymbol{b}_{\mathcal{J}}$ indicates that only those bits are selected from $\boldsymbol{b}$ that are indicated in $\mathcal{J}$). Finally, X and X' are used for the construction of a secret key, e.g. using the algorithm described in Section 2.2.

An analysis of error probabilities and key lengths is presented in Appendix A. It turns out (see Eqs. 5,6) that the bit error probability in X' is drastically improved compared to the 'one way' case, where only the enrolled helper data is used ($X_{\mathrm{1way}} = \boldsymbol{b}_{\mathcal{I}}$; $X'_{\mathrm{1way}} = \boldsymbol{b}'_{\mathcal{I}}$). As a consequence, the amount of computational effort spent on the error correction using $\mathcal{E}$ is greatly reduced (linear in the number of correctible errors). Furthermore, it turns out that the extracted keys are longer because fewer redundancy bits are needed (see Eq. 8). For a reasonable choice of parameters, the improvement in bit error probability in X' can be as small as a factor 5 and as large as 50. The simultaneous improvement in key length varies between 20% and 70%. The difference between the two methods is most pronounced when the measurements are very noisy.

3 Noise reduction for optical PUFs

3.1 'Pyramid' structure

In Fig. 1 we present an elegant way of detecting misalignments between an optical PUF and a camera. At the bottom of the PUF, a small pyramid-shaped volume is removed. When the laser beam enters the PUF, a fraction of the light reaches the bottom without being scattered by the random particles. There a certain fraction reflects off the pyramid structure and is divided into four sub-beams. These beams are partially transmitted through the PUF without scattering, and give rise to four bright spots on the camera. The spots are superimposed on the speckle pattern. Misalignments (translations and rotations in all directions) can be uniquely read off from the relative positions of the four spots (see Fig. 1 a–d). This allows the reader to adjust its settings.

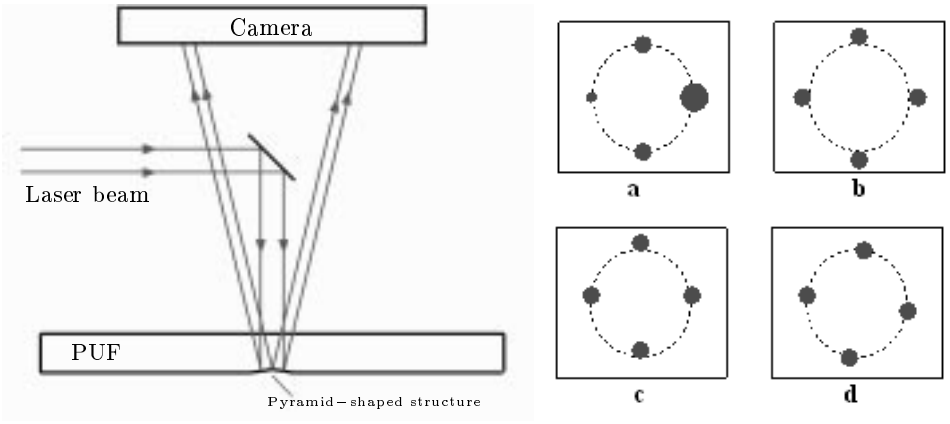

Fig. 1. Left: Light scattering from the pyramid structure. **Right:** Effects of misalignment. (a) Shift in x-direction. (b) Shift in z-direction. (c) Rotation around the x-axis. (d) Rotation around the z-axis.

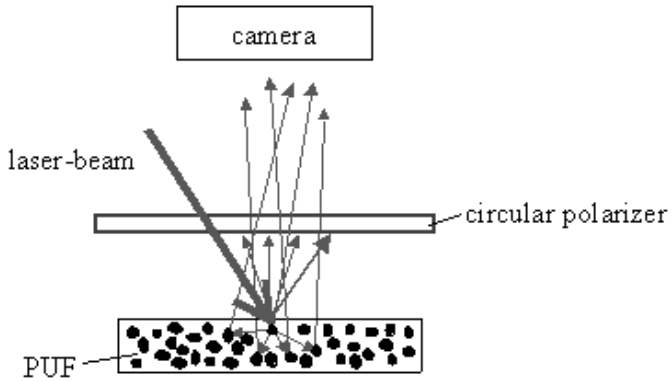

Fig. 2. *Circular polariser blocking light that reflects directly from the top of the PUF.*

3.2 Polarisation selection

The noise due to scratches and dirt on the surface of an optical PUF can be reduced by making use of the fact that light changes its polarisation when it is reflected. The method works as follows. We assume a geometry as in Fig. 2. When the laser light is generated, it has *linear* polarisation. On its way to the PUF the beam passes through a *circular* polariser. Light that gets scattered from the top of the PUF, without entering it, will have reversed circular polarisation and hence will be absorbed when it meets the polariser again. Light that enters the PUF, however, is subjected to multiple scattering, which has a depolarising effect. Hence, a substantial fraction of the multiply scattered light will pass the polariser and reach the camera. In this way, direct reflection from scratches and dirt is eliminated. In order to improve the selectivity, one can add an additional quarter wave plate on top of the PUF; passing through it twice precisely negates the polarisation-reversing effect of a reflection.

4 Experimental results for optical PUFs

We show experimental results that demonstrate the effectiveness of helper data in the form of robust components. The algoritm of Section 2.2 was applied, without making use of the techniques decribed in Sections 2.3, 2.4 and 3. We used the following setup. The laser is a DBF laser with a wavelength of 785 nm (spectral width 1nm). The beam diameter is 1 mm. We have used five scattering samples with a thickness of 0.4 mm. Pictures of the reflected speckle pattern are taken with a 1024 by 768 pixel CCD camera with a pixel pitch of 6.25 mm. The bitmap has 256 gray levels. The distance between the laser and the sample is 10 cm, and the distance from the sample to the camera is 13 cm.

4.1 Binarized Gabor coefficients

In order to extract bit strings from speckle images we have used the method of Gabor Transforms as proposed in [3]. Gabor Transforms are well suited since they are insensitive to small changes in an image and they reveal the locations as well as the orientations of structures at different spatial frequencies. They are used in a wide range of applications, such as iris recognition [17], texture analysis and image enhancement, coding and compression.

A two-dimensional Gabor basis function $\Gamma(s, \boldsymbol{k}, \boldsymbol{x}_0, \boldsymbol{x})$ is the product of a plane wave with wave vector $\boldsymbol{k}$ and a Gaussian centered on $\boldsymbol{x}_0$ with width s. ($\boldsymbol{x}$ denotes a location in the speckle image). We write the Gabor basis functions Γ and the Gabor coefficients G as follows.

$$G_{\mathrm{IM}}(s, \boldsymbol{k}, \boldsymbol{x}_0) = \int \mathrm{d}^2x \; \Gamma_{\mathrm{IM}}(s, \boldsymbol{k}, \boldsymbol{x}_0, \boldsymbol{x}) I(\boldsymbol{x}) \tag{1}$$

$$\Gamma_{\mathrm{IM}}(s, \boldsymbol{k}, \boldsymbol{x}_0, \boldsymbol{x}) = \frac{1}{s\sqrt{2\pi}} \sin \boldsymbol{k} \cdot (\boldsymbol{x} - \boldsymbol{x}_0) \exp[-\frac{(\boldsymbol{x} - \boldsymbol{x}_0)^2}{4s^2}]. \tag{2}$$

Here I denotes the light intensity. We have selected the imaginary part of the transform, since it is invariant under spatially constant shifts of I. In the notation of Section 2, a bitmap image of a speckle pattern corresponds to the 'raw' bit-string $\boldsymbol{b}$. The 'robust' bit-string X is obtained as follows. Gabor coefficients G_{IM} are evaluated for a set of parameters s, $\boldsymbol{k}$, $\boldsymbol{x}_0$. Coefficients are discarded if they do not exceed a certain threshold T, i.e. one only keeps $|G_{\mathrm{IM}}| > T$. Finally, the robust coefficients are binarized; positive values are mapped to '1' and negative to '0'.

Attention must be paid to the fact that Gabor coefficients can be strongly correlated. Ideally one should construct a bit-string from values that are almost independent. In general, correlations between $G_{\mathrm{IM}}(s, \boldsymbol{k}, \boldsymbol{x}_0)$ and $G_{\mathrm{IM}}(s', \boldsymbol{k}', \boldsymbol{x}_0')$ occur when their parameters do not differ much. Correlations also occur if $|\boldsymbol{x}_0' - \boldsymbol{x}_0|$ is smaller than the speckle size. An analysis of these correlations is presented in Appendix B. For simplicity we have used the following parameters in our experiments: A single Gaussian width $s = 13$ pixels, a single length $|\boldsymbol{k}| = \pi/8$ pixels^{-1}, two directions of $\boldsymbol{k}$ (45° and 135°), and $\boldsymbol{x}_0$ positions in a square grid with a spacing of 8 pixels. This yields 2400 Gabor coefficients. There are very strong correlations (≈ 0.9) between diagonal neighbours on the $\boldsymbol{x}_0$-grid when $\boldsymbol{k} \parallel \boldsymbol{k}'$ and $(\boldsymbol{x}_0' - \boldsymbol{x}_0) \perp \boldsymbol{k}$. Furthermore, there are strong anti-correlations (≈ -0.7) between diagonal neigbours when $\boldsymbol{k}$, $\boldsymbol{k}'$ and $(\boldsymbol{x}_0' - \boldsymbol{x}_0)$ point in the same direction. Other correlations are zero or negligible. This explains the stripes in Fig. 3.

The robustness threshold T was chosen such that in the enrollment phase there are always more than 1023 Gabor coefficients exceeding the threshold. We have used a BCH code with parameters (1023, 56, 191), i.e. 1023-bit code words, 56-bit message words (the actual key length), and correction of 191 errors. The high error-correcting capacity is necessary because the bit error rate (BER) in the robust bit-string X' is still high when no special measures are taken to reduce the noise. Without showing proof we mention that the Calibration CRP method reduces the BER to $< 5\%$, allowing for a BCH code with parameters (1023,553,52), i.e. robust 553-bit message words. Note, however, that the actual information content (entropy) is lower than 553 bits due to the strong correlations between the Gabor coefficients (see Appendix B).

The statistics of the Gabor coefficients is the subject of ongoing research.

Fig. 3. Left: *Example of a speckle pattern.* **Middle:** *Binarized Gabor coefficients at 45°.* **Right:** *Binarized Gabor coefficients at 135°.*

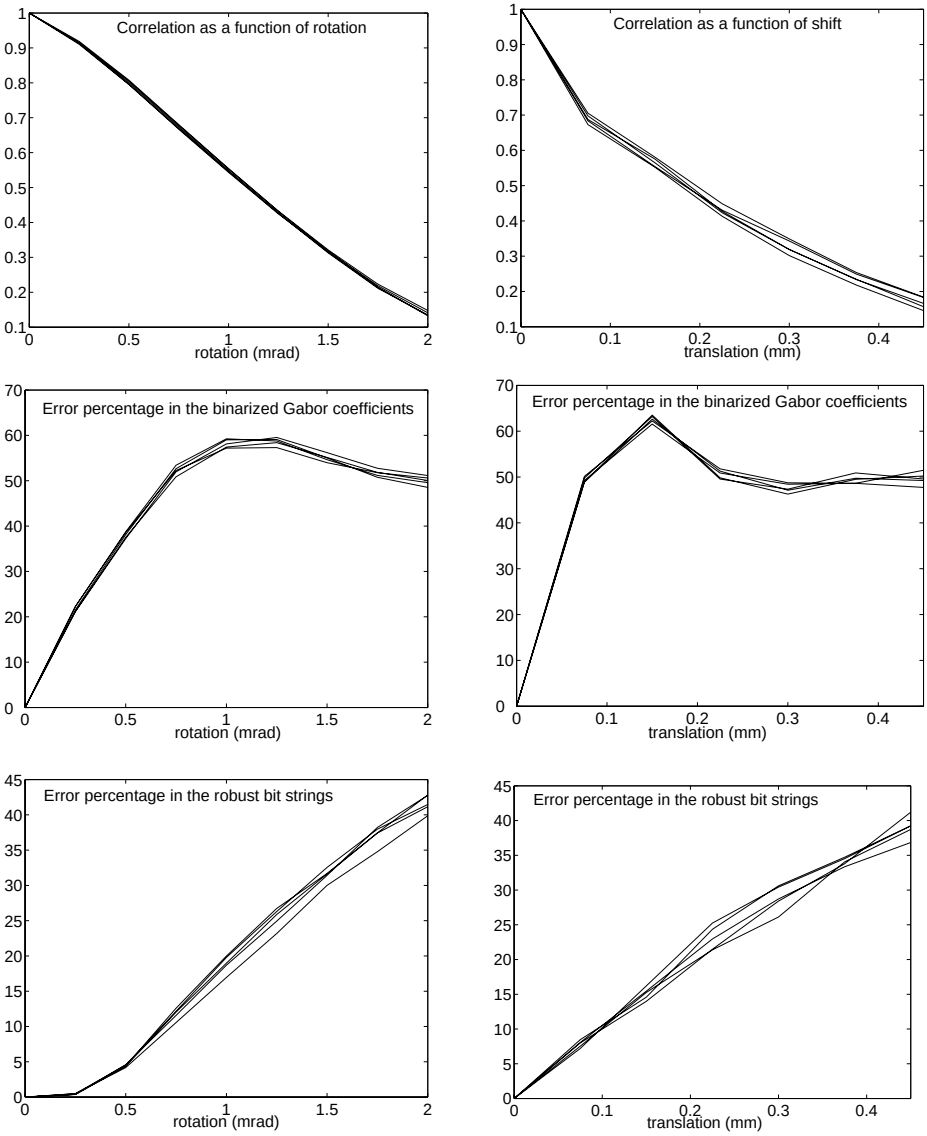

Fig. 4. *Effects of misalignment. The five curves correspond to different samples.* **Left:** *Tilting of the laser beam.* **Right:** *Shift of the sample.* **From top to bottom:** *Correlation between original and perturbed speckle pattern; Error percentage in the binarized Gabor coefficients (2400-bit string); Error percentage in the selected robust bit-string (>1023 bits).*

4.2 Experimental results

Fig. 3 shows a typical speckle pattern and the binarized Gabor coefficients. We studied the sensitivity of the binarized coefficients as well as the selected robust coefficients under small rotations and translations. All measurements were repeated ten times (re-inserting the samples each time) and averaged over these ten instances. As a direct measure of the difference between two speckle patterns B_1,B_2 we use the correlation $C_{\text{bmp}} \in [-1, 1]$ between the bitmaps,

$$C_{\text{bmp}} = \frac{\langle B_1(\boldsymbol{x}_i) B_2(\boldsymbol{x}_i) \rangle_i - \langle B_1(\boldsymbol{x}_i) \rangle_i \langle B_2(\boldsymbol{x}_i) \rangle_i}{\sigma_1 \sigma_2} \tag{3}$$

where $\langle \cdot \rangle_i$ denotes the spatial average and σ is the standard deviation in the gray level of the speckle pattern. The results of the measurements are shown in Fig. 4. The graphs show that for rotations larger than 0.7 mrad and shifts larger than 0.1 mm, the binarized coefficients look completely independent (50% errors). The robust bits, however, are significantly more resilient: There the BER level of 50% is reached only at rotations $>$ 2mrad and shifts $>$ 0.5mm. This demonstrates the usefulness of robust components as a form of helper data.

ACKNOWLEDGEMENTS
We thank Marten van Dijk, Vincent van der Leest, Sjoerd Stallinga and Ton Akkermans for useful discussions.

References

1. K.M. Tolk, *Reflective Particle Technology for Identification of Critical Components*, 33rd Annual Meeting Proceedings of the Institute of Nuclear Materials Management, July 1992.

2. Unicate BV's '3DAS' system, `http://www.andreae.com/Unicate/Appendix%201.htm`, 1999.

3. R. Pappu, *Physical One-Way Functions*, Ph.D. thesis, MIT 2001.

4. R. Pappu, B. Recht, J. Taylor, N. Gershenfeld, *Physical One-Way Functions*, Science Vol. 297, p.2026, Sept 2002.

5. P. Tuyls, B. Škorić, S. Stallinga, A.H.M. Akkermans, W. Ophey, *Information-Theoretic Security Analysis of Physical Uncloneable Functions*, Proc. Financial Cryptography and Data Security 2005.

6. B. Gassend, D. Clarke, M. van Dijk, S. Devadas, *Controlled Physical Random Functions*, Proc. 18th Annual Computer Security Applications Conf., Dec. 2002.

7. P. Tuyls, B. Škorić, *Secret Key Generation from Classical Physics*, Proceedings of the Hardware Technology Drivers for Ambient Intelligence Symposium, Philips Research Book Series, Kluwer, 2005.

8. B. Gassend, D. Clarke, M. van Dijk, S. Devadas *Silicon Physical Random Functions*, Proc. 9th ACM Conf. on Computer and Communications Security, 2002.

9. B. Gassend, *Physical Random Functions*, Master's Thesis, MIT 2003.

10. M. Magnor, P. Dorn, W. Rudolph, *Simulation of confocal microscopy through scattering media with and without time gating*, J.Opt.Soc.Am. B, Vol. 19, no. 11 (2001), 1695–1700.

11. J. F. de Boer, *Optical Fluctuations on the Transmission and Reflection of Mesoscopic Systems*, Ph.D. thesis, 1995, Amsterdam.

12. H. Furstenberg, *Noncommuting Random Matrices*, Trans. Am. Math. Soc. 108, 377, 1963.

13. P. Tuyls, J. Goseling, *Capacity and Examples of Template Protecting Biometric Authentication Systems*, Biometric Authentication Workshop (BioAW 2004), LNCS 3087, 158–170, Prague, 2004.

14. Y. Dodis, L. Reyzin, A. Smith, *Fuzzy Extractors: How to generate strong secret keys from biometrics and other noisy data*, in Advances in Cryptology – Eurocrypt'04, LNCS 3027, 523–540, 2004.

15. A. Juels, M. Wattenberg, *A Fuzzy Commitment Scheme*, in G. Tsudik, ed., Sixth ACM Conference on Computer and Communications Security, 28–36, ACM Press. 1999.

16. J.P. Linnartz, P. Tuyls, *New Shielding Functions to enhance Privacy and Prevent Misuse of Biometric Templates*, Proc. 4th International Conference on Audio and Video based Biometric Person Authentication, LNCS 2688, Guildford UK, June 9-11, 2003.

17. J. Daugman, *The importance of being random; statistical principles of iris recognition*, Pattern Recognition 36, 279–291, 2003.

18. J. W. Goodman, *Statistical properties of laser speckle patterns*, in Laser Speckle and Related Phenomena, 2nd ed., J. C. Dainty, Ed. New York: Springer-Verlag, 1984.

A Two-way use of helper data

In this Appendix we use a simple model to analyse the effects of using the helper data that is generated during the verification phase as proposed in Section 2.4. We consider the measurement of n variables $x_1, \ldots, x_n$, representing the PUF response, which are independent and identically distributed according to a normal distribution with zero mean and standard deviation Σ_x. (This is sometimes called the 'inter-class' variation). The measurement error due to misalignment and external noise is assumed to be independently Gaussian distributed with

standard deviation σ ('intra-class' variation). If the enrollment measurement yields a value f, with absolute value larger than some threshold T, the value is deemed 'robust'. We compute the probability $P_{\rm robust}$ of finding a robust value when a noisy measurement is done of a variable x_i, given that the 'noiseless' value of x_i is *unknown*. We have to take the inter-class variation into account and hence average over x_i,

$$P_{\rm robust} = 1 - \int_{-T}^{T} {\rm d}f \int_{-\infty}^{\infty} {\rm d}x \; N_{0\Sigma_x}(x) N_{x\sigma}(f) = 1 - {\rm Erf} \frac{T}{\sqrt{2}\sqrt{\Sigma_x^2 + \sigma^2}}. \qquad (4)$$

Here the notation $N_{\mu s}$ stands for the normal distribution with mean μ and standard deviation s, and Erf denotes the Error Function. Given a robust measured f, the probability P_1 that a bit flip will occur in the second measurement, according to the one-way method, is equal to the probability that the second measurement yields a number F with sign opposite from f. Taking $f > 0$ without loss of generality, this probability is

$$P_1 = \int_{-\infty}^{\infty} {\rm d}x \; N_{f\sigma}(x) \int_{-\infty}^{0} {\rm d}F \; N_{x\sigma}(F) = \frac{1}{2} - \frac{1}{2} {\rm Erf} \frac{f}{2\sigma}. \qquad (5)$$

The first integral in (5) is an average over all the possibilities for the unknown 'true' value x. Given the fact that f was obtained in the first measurement, x is Gaussian-distributed around f, with standard deviation given by the noise strength σ.

On the other hand, if the two-way helper data method is used, the probability of a bit flip (P_2) is equal to the probability that F not only has opposite sign, but also has absolute value larger than the threshold T,

$$P_2 = \int_{-\infty}^{\infty} {\rm d}x \; N_{f\sigma}(x) \int_{-\infty}^{-T} {\rm d}F \; N_{x\sigma}(F) = \frac{1}{2} - \frac{1}{2} {\rm Erf} \frac{f+T}{2\sigma}. \qquad (6)$$

The amount of computational effort that has to be spent on error-correcting codes is roughly linear in the expected number of errors. Hence we are interested in the expectation values $\langle P_1 \rangle$ and $\langle P_2 \rangle$, where the brackets denote averaging with respect to f (with $f \geq T$). Making a natural choice for the parameters, $\sigma < T < 2\sigma$ and $\Sigma_x > \sigma$, it turns out that the ratio $\langle P_1 \rangle \, / \, \langle P_2 \rangle$ lies in a range between approximately 5 and 50 (increasing with T/σ), indicating that the two-way method gives a huge reduction of the computational cost of using the error-correcting code $\mathcal{E}$.

One may worry that the two-way method yields shorter keys, as more bits are being discarded in the establishment of the robust bit-string X'. We show that, on the contrary, longer keys are extracted. In the one-way method, a variable x_i that has been found to be robust at enrollment ($f > T$) is always kept. In the two-way method there is a nonzero probability $P_{\rm discard}$ of discarding such a variable,

$$P_{\rm discard} = \int_{-\infty}^{\infty} {\rm d}x \; N_{f\sigma}(x) \int_{-T}^{T} {\rm d}F \; N_{x\sigma}(F) = \frac{1}{2} {\rm Erf} \frac{f+T}{2\sigma} - \frac{1}{2} {\rm Erf} \frac{f-T}{2\sigma}. \qquad (7)$$

We denote the length of the robust string X' in the one-way method as $n_1 = n{\cdot}P_{\rm robust}$. The corresponding length in the two-way case is $n_2 = n_1(1-\langle P_{\rm discard}\rangle)$, i.e. shorter than n_1. However, it is well known that the information capacity of a channel strongly depends on the error rate of the channel. Given an error rate p, the information content per transmitted bit is $1 - h(p)$, with $h(p) = -p\log p - (1-p)\log(1-p)$. The maximum entropy H of the derived key K in the two methods is given by

$$H_1 = n_1[1 - h(\langle P_1\rangle)]\ ; \qquad H_2 = n_2[1 - h(\langle P_2\rangle)]. \tag{8}$$

For given signal to noise ratio Σ_x/σ, an optimal choice of T/σ exists (for each method separately) that yields the highest entropy. It turns out that the best H_2 is always larger than the best H_1. The difference between the two methods is most pronounced at small Σ_x/σ, i.e. noisy measurements.

B Correlations between Gabor Coefficients

In this Appendix we compute the correlation between the Gabor coefficients (1). We use the shorthand notation $G_{\rm IM} = G_{\rm IM}(s, \boldsymbol{k}, \boldsymbol{x})$ and $G'_{\rm IM} = G_{\rm IM}(s', \boldsymbol{k}', \boldsymbol{x}')$. By σ_G and σ'_G we denote the standard deviation of $G_{\rm IM}$ and $G'_{\rm IM}$ respectively. We define the correlation $C_{\rm G} \in [-1, 1]$ as

$$C_{\rm G} := \frac{\langle G_{\rm IM} G'_{\rm IM}\rangle - \langle G_{\rm IM}\rangle\,\langle G'_{\rm IM}\rangle}{\sigma_G \sigma'_G} = \frac{\langle G_{\rm IM} G'_{\rm IM}\rangle}{\sqrt{\langle (G_{\rm IM})^2\rangle\,\langle (G'_{\rm IM})^2\rangle}}. \tag{9}$$

The brackets denote averaging over speckle patterns. For the last equality we have used the fact that $\mathit{\Gamma}_{\rm IM}$ (2) is an odd function of $\boldsymbol{x}$, which leads to $\langle G_{\rm IM}\rangle = 0$ regardless of the choice of parameters. For the computation of the expectation values we use a result from [18],

$$R(\boldsymbol{x}_1, \boldsymbol{x}_2) := \langle I(\boldsymbol{x}_1) I(\boldsymbol{x}_2)\rangle = 4\left[\frac{J_1(|\boldsymbol{x}_2 - \boldsymbol{x}_1|/M)}{|\boldsymbol{x}_2 - \boldsymbol{x}_1|/M}\right]^2 \tag{10}$$

where J_1 is a Bessel function and M is a constant proportional to the average speckle size, $M = \lambda z/(2\pi W)$, with λ the wavelength, z the distance between the exit plane of the PUF and the detector, and W the diameter of the illuminated area of the PUF. Substitution of (10) and (2) into (9) gives

$$\frac{\langle G_{\rm IM} G'_{\rm IM}\rangle}{\langle I\rangle^2} = \int \frac{\mathrm{d}^2 x_1 \mathrm{d}^2 x_2}{2\pi s s'} R\; e^{-\frac{(\boldsymbol{x}_1 - \boldsymbol{x})^2}{4s^2} - \frac{(\boldsymbol{x}_2 - \boldsymbol{x}')^2}{4s'^2}} \sin\boldsymbol{k}\cdot(\boldsymbol{x}_1 - \boldsymbol{x}) \sin\boldsymbol{k}'\cdot(\boldsymbol{x}_2 - \boldsymbol{x}'). \tag{11}$$

We introduce 'center of mass' coordinates as follows,

$$\begin{aligned}
\boldsymbol{x} &= \bar{\boldsymbol{x}} - \tfrac{1}{2}\boldsymbol{\Delta} \quad ; \quad \boldsymbol{x}' &&= \bar{\boldsymbol{x}} + \tfrac{1}{2}\boldsymbol{\Delta} \\
\boldsymbol{x}_1 &= \boldsymbol{m} - \tfrac{1}{2}\boldsymbol{\delta} \quad ; \quad \boldsymbol{x}_2 &&= \boldsymbol{m} + \tfrac{1}{2}\boldsymbol{\delta} \\
\boldsymbol{k} &= \boldsymbol{K} - \tfrac{1}{2}\boldsymbol{\zeta} \quad ; \quad \boldsymbol{k}' &&= \boldsymbol{K} + \tfrac{1}{2}\boldsymbol{\zeta} \\
1/s^2 &= p - \tfrac{1}{2}q \quad ; \quad 1/s'^2 &&= p + \tfrac{1}{2}q
\end{aligned} \tag{12}$$

In terms of these coordinates, the expectation value (11) can be expressed as

$$\begin{aligned}\langle G_{\rm IM} G'_{\rm IM}\rangle = \frac{\langle I\rangle^2}{\pi s s'} \int {\rm d}^2\delta \ \left[\frac{J_1(\delta/M)}{\delta/M}\right]^2 \exp[-\tfrac{p}{8}(\boldsymbol{\delta}-\boldsymbol{\Delta})^2] \\ \int {\rm d}^2 m \ \exp[-\tfrac{p}{2}m^2 - \tfrac{q}{4}\boldsymbol{m}\cdot(\boldsymbol{\delta}-\boldsymbol{\Delta})] \\ \times \left\{\cos[\boldsymbol{K}\cdot(\boldsymbol{\delta}-\boldsymbol{\Delta}) + \boldsymbol{\zeta}\cdot\boldsymbol{m}] - \cos[2\boldsymbol{K}\cdot\boldsymbol{m} + \tfrac{1}{2}\zeta\cdot(\boldsymbol{\delta}-\boldsymbol{\Delta})]\right\}. \end{aligned} \tag{13}$$

Here we have assumed, without loss of generality, that $\bar{\boldsymbol{x}} = \boldsymbol{0}$. The m-integral is readily evaluated, yielding

$$\begin{aligned}\langle G_{\rm IM} G'_{\rm IM}\rangle = \frac{2\,\langle I\rangle^2}{p s s'} \int {\rm d}^2\delta \ \left[\frac{J_1(\delta/M)}{\delta/M}\right]^2 \exp[-(\tfrac{p}{8} - \tfrac{q^2}{32p})(\boldsymbol{\delta}-\boldsymbol{\Delta})^2] \\ \times \left\{ e^{-(1/2p)\zeta^2} \cos[(\boldsymbol{K} - \tfrac{q}{4p}\boldsymbol{\zeta})\cdot(\boldsymbol{\delta}-\boldsymbol{\Delta})] \right. \\ \left. - e^{-(2/p)K^2} \cos[(\tfrac{1}{2}\boldsymbol{\zeta} - \tfrac{q}{2p}\boldsymbol{K})\cdot(\boldsymbol{\delta}-\boldsymbol{\Delta})]\right\}. \end{aligned} \tag{14}$$

The δ-integral cannot be evaluated analytically. Several trends can be observed, however. The integrand contains a rapidly decreasing function of $\boldsymbol{\delta}$ centered around $\delta = 0$, with scale M, times another rapidly decreasing function of $\boldsymbol{\delta}$ centered around $\boldsymbol{\Delta}$, with scale $\approx s$. Hence, if Δ is larger than $\min(M, s)$, then the expectation value (14) becomes very small. Furthermore, it can also be seen that the δ-integral becomes small when $\zeta^{-1} \ll \min(M, s)$, because then the oscillations cancel each other.

We make an approximation by writing $4[J_1(u)/u]^2 \approx \exp(-u^2/2\Sigma^2)$, with $\Sigma \approx 1.29$. This makes the δ-integral manageable and nicely captures the decay of the integrand between $u = 0$ and $u \approx 3.83$ where $J_1(u) = 0$, but the asymptotic behaviour at large u is misrepresented. Hence, the approximation is useful for small Δ. We present the result for $s' = s$:

$$\begin{aligned} C_{\rm G}(s' = s) \approx \exp\left[-\frac{1}{2}\cdot\frac{\Delta^2}{M^2\Sigma^2 + 4s^2}\right] \times \\ \frac{e^{\Gamma s^2 \boldsymbol{k}\cdot\boldsymbol{k}'} \cos\tfrac{\Gamma}{2}\boldsymbol{\Delta}\cdot(\boldsymbol{k}'+\boldsymbol{k}) - e^{-\Gamma s^2\boldsymbol{k}\cdot\boldsymbol{k}'}\cos\tfrac{\Gamma}{2}\boldsymbol{\Delta}\cdot(\boldsymbol{k}'-\boldsymbol{k})}{2\sqrt{\sinh\Gamma s^2 k^2}\sqrt{\sinh\Gamma s^2 k'^2}} \end{aligned} \tag{15}$$

where $\Gamma \in [0,1]$ is defined as $\Gamma = [1 + M^2\Sigma^2/(4s^2)]^{-1}$.