MEMS-based Gyroscopes as Physical Unclonable Functions

Oliver Willers*, Christopher Huth†, Jorge Guajardo‡ and Helmut Seidel§
*†Research and Advance Engineering, Robert Bosch GmbH, Stuttgart, Germany
Email: *Oliver.Willers@de.bosch.com, †Christopher.Huth@de.bosch.com
‡Research and Technology Center, Bosch LLC, Pittsburgh, USA
Email: Jorge.GuajardoMerchan@bosch.com
§Chair of Micromechanics, Microfluidics/Microactuators, Saarland University
Email: seidel@lmm.uni-saarland.de

***Abstract*—We are at the dawn of a hyper connectivity age otherwise known as the Internet of Things (IoT). It is widely accepted that to be able to reap all benefits from the IoT promise, device security will be of paramount importance. A key requirement for most security solutions is the ability to provide secure cryptographic key storage in a way that will easily scale in the IoT age. In this paper, we focus on providing such a solution based on Physical Unclonable Functions (PUFs). To this end, we focus on microelectromechanical systems (MEMS)-based gyroscopes and show via wafer-level measurements and simulations, that it is feasible to use the physical and electrical properties of these sensors for cryptographic key generation. After identifying the most promising features, we propose a novel quantization scheme to extract bit strings from the MEMS analog measurements. We provide upper and lower bounds for the minimum entropy of the bit strings derived from the measurements and fully analyze the intra- and inter-class distributions across the operation range of the MEMS device. We complement these measurements via Monte-Carlo simulations based on the distributions of the parameters measured on actual devices. We also propose and evaluate a key derivation procedure based on fuzzy extractors for Hamming distance, using the min-entropy estimates obtained to derive a full entropy 128-bit key, requiring 1219-bits of helper data with an (authentication) failure probability of $4 \cdot 10^{-7}$. Thereby, we present a complete cryptographic key generation chain. In addition, we propose a dedicated MEMS-PUF design, which is superior to our measured sensor, in terms of chip area, quality and quantity of key seed features.**

I. Introduction

In 1991, Mark Weisser [1] set out the vision of ubiquitous computation, which promised to make our interaction with things to be seemless. Today, this vision has already started to become reality through modern technologies that allow for electronic systems to be embedded practically everywhere with applications ranging from smart homes, to connected vehicles and smart factories. More specifically, ubiquitous computation has been made tangible in the concept of the Internet of Things (IoT), which by some estimates is expected to surpass 50 billion devices by 2020 [2]. Regardless of the exact numbers, it is widely acknowledged that to make the IoT a success the security of this super large distributed systems will have to be guaranteed and the privacy of the collected data protected.

The Internet of Things, made possible through the wide deployment of embedded devices, differs significantly from ”classical” systems, such as desktop (networked) PCs, in various aspects, which include: severe computational, memory, and power constraints, lack of advanced user interfaces, an increased vulnerability with respect to physical or network attacks, and as mentioned previously, their tendency to collect potentially highly privacy sensitive data. Until recently, there has been an inclination to assume the inability to provide strong hardware security guarantees. However, this is starting to change with new device architectures such as those presented in [3]–[5], which aim to provide more fundamental security properties for embedded devices. In this paper, we continued this line of work and we focus our attention on an even more constrained type of device, MEMS-based sensor devices, which are widely deployed today in smart phones, automotive applications (e.g., crash detection, airbag deployment), environmental condition assessment, pressure measurements, etc. and for which security solutions have been until now overlooked.

As a starting point in the study security for MEMS-based sensors, we look at how to provide secure cryptographic key storage in such devices in a cheap and intrinsic manner, as keeping cryptographic keys secure is the basis for many higher level security mechanisms such as attestation, secure boot as well as any other cryptographic operation which might require a secret or private key (e.g., encryption, signatures, message authentication generation, etc.). In particular, we look at the feasibility of creating a Physical Unclonable Function (PUF) based on the physical properties of MEMS devices themselves. PUFs have received a lot of attention (see e.g., [6]–[10]) as a technology for secure key storage. One of PUF’s main advantages is that the device does not need to store secrets in non-volatile memory but rather it can generate the cryptographic key whenever it needs to process secrets and destroys it afterward, making the job of an attacker with physical access to the device more difficult[1].

While the possibility of deriving a fingerprint from MEMS-based devices has been explored in previous work [14], the feasibility of deriving a cryptographic key from MEMS char-

[1] The fact that memory is susceptible to invasive attacks has been demonstrated in [11]–[13].

acteristics is a more challenging undertaking and to the best of our knowledge, we are the first to propose such a design. As with many PUFs, a MEMS-based PUF has the following requirements: the cryptographic key should be unique per device (similar to a fingerprint), (ii) the cryptographic key should be reproducible across the whole range of environmental conditions for which the device is designed, (iii) the cryptographic key should be hard to replicate even for the manufacturer of the device, (iv) the PUF properties should be hard to model and therefore a mathematical model that predicts the PUF responses should be infeasible to obtain, and (v) it is desirable that the particular PUF has tamper resistance or tamper evidence properties. In this paper, we show that MEMS-based gyroscopes can be used to this end and, moreover, we show via experimental evidence on actual devices and simulations that requirements (i)-(iv) are met by our design. Furthermore, we present and simulate a fully functional MEMS device specifically designed for PUF applications, which has smaller size than other gyroscopes and has more variation (allowing for the derivation of more full entropy bits). In short, our contributions are as follows:

- **Physical Modelling**: In contrast to previous work, which use the response of MEMS accelerometers and derive signal processing features suitable for identification, we identify suitable properties (mechanical and electrical) of the MEMS gyroscopes and show that they can be used to derive a robust bit string suitable for cryptographic key generation,
- **Key Derivation**: We propose a quantization method which allows us to derive binary keys from analog sensor data inspired by a method described by Chang et al. [15]. Then, we analyze via multiple methods the amount of entropy that such binary strings carry and based on a conservative estimate we propose several helper data [16], [17] parameters which would provide with robust keys across a temperature range of 65 °C, with probabilities of failure lower than 10^{-6}. We also provide specific codes, which can be used in combination with a fuzzy extractor to create a uniformly distributed random 128-bit key.
- **Uniqueness and Robustness**: We analyze the intra- and inter-class distributions induced by our key derivation procedure from 70 different physical MEMS-devices and verify the behavior of such distributions via Monte-Carlo simulations of the MEMS behavior using variability parameters measured on physical MEMS devices. This analysis includes the variability due to repeated measurements and environmental conditions, most prominantly, temperature.
- **MEMS Design Optimized for PUF Applications**. We present a completely new MEMS design, which has been optimized to increase variability and thus, the ability to create unique/robust keys

A. Organization of the Paper

We begin by providing basic background on MEMS technology, their potential for PUFs and cause of variations in Section II. In Section III, we show how a MEMS-PUF should be included in a package, to withstand probing attacks. We then explain features of MEMS that fulfill our requirements for robustness and uniqueness in Section IV, how we quantize these features, how our measurements are set up and the results for the most promising parameters. From the learned insights, we then can simulate additional devices in Section V. This allows us to verify that the simulations are consistent with the measured data. In Section VI, we provide upper and lower bounds for the min-entropy of the MEMS-PUF responses for both measured and simulated data. In Section VII, we describe the last step in the key generation process, namely, information reconciliation via error correcting codes and randomness extraction. It is worth observing that our constructions tend to require less public helper data (measured in bits) than recently published fuzzy extractor schemes, in spite of our constructions are based on very conservative min-entropy estimations[2]. We propose a dedicated MEMS-PUF design in Section VIII. We conclude this article in Section IX.

II. MEMS Background

MEMS sensors are silicon based devices which combine a microcontroller with a mechanical device used to measure a variety of different physical quantities ranging from acceleration and yaw rate to magnetic fields, pressure, humidity, etc. In this work, we focus on MEMS-based gyroscopes which are devices for measuring the yaw rate. MEMS-based gyroscopes are very complex entities with a large number of mechanical as well as electrical properties. A MEMS gyroscope typically consists of a combination of one or several spring-mass systems which oscillate at resonant frequency. In order to drive the system, an external source is needed that applies the required voltage. To detect the yaw rate, the Coriolis effect is used. This effect is based on the Coriolis force, which acts on a moved mass in a rotating system. The Coriolis force causes a deflection of an oscillating mass which is proportional to the acting yaw rate. Therefore, the yaw rate can be determined by measuring this deflection in a capacitive way. The detecting axis depends on the moving direction of an oscillating mass. For each detecting axis, at least one oscillating spring-mass system is needed. This means that the number of different spring-mass systems depends basically on the number of sensitive axis. In this work, a 3-channel gyroscope was investigated. For further background on gyroscopes we refer the reader to [18].

A. MEMS Parameters Suitable for Identification

MEMS sensors offer many measurable mechanical as well as electrical, parameters depending on the sensor type, which can be used to derive a suitable unique identifier and, after

[2] In the PUF literature, it is standard to use the Context Tree Weighing (CTW) compression algorithm to estimate entropy of the PUF responses. We use CTW as an upper bound on the entropy of the MEMS-PUF responses but use the more conservative min-entropy estimations provided by the NIST tests for our final helper data sizes.

some processing, a secure cryptographic key. In the case of MEMS-based gyroscopes, fundamental mechanical parameters include the different frequency modes of the sensor. MEMS-based gyroscopes have a complex mechanical structure which consists of several spring-mass systems. Hence, a large number of frequency modes exist for MEMS-based gyroscopes. Another interesting mechanical parameter is the quadrature which is a measure for the asymmetry of a sensor. As the manufacturing process is subjected to variations, the actual physical structures, i.e., springs, masses and electrode gaps, differ slightly from the ideal case by different types of asymmetries. This can result in a deflection of the moving directions and produces an error signal called the quadrature signal - which is detected by electrodes in a capacitive manner. Additionally, there are a lot of electrical parameters. These are the capacitances and resistances that are induced between the different electrodes which are needed for driving and measuring the sensor. Other properties are the ability for frequency tuning, quality factors and decay times. However, we do not describe them any further because they have not proven to be suitable as PUF parameters in our evaluation.

B. Causes for Parameter Variability

Although, it is difficult to determine *all* influencing factors affecting the silicon manufacturing process, several of them are well-known and understood. In what follows, we provide an overview of the fundamental factors and their impact on parameter variation. A main factor for the parameter variability is the variation of the geometric dimensions (width and thickness of the structures) that occurs always in the etching process and it varies in a small range. This includes a variation of the beam width of the springs and, hence, it changes the spring rigidity, which leads to a shift of the resonant frequencies. In addition, it affects the electrical parameters as well because it changes the gaps between the electrodes and the effective area of electrodes.

As mentioned previously, asymmetries cause slight variations of the behavior from the ideal case generating the quadrature signal. These asymmetries have four sources:

1) A difference of the side wall inclinations, causing a different deviation from the rectangular beam geometry of side walls that results in an undesirable out-of-plane force component.
2) A local variation of the structure width, affecting slightly the spring rigidities.
3) An imbalance of the inertial masses.
4) The influence of mechanical stress caused by packaging, temperature and bending of the Printed Circuit Board (PCB) after soldering.

Note that actual MEMS sensors are designed with the objective of minimal parameter variations. In principle, an amplification of the parameters' variation is easy to achieve. Notice that such an amplication is likely to result in an increase in the number of bits extracted from a particular parameter. This could be used for the creation of a dedicated MEMS structure to increase significantly the number of bits that can be derived.

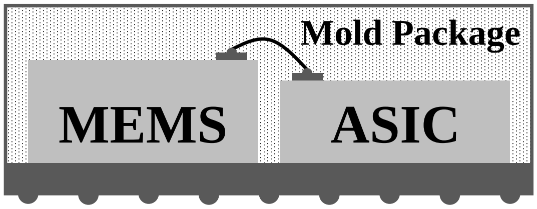


Fig. 1. Schematic composite of MEMS sensor and ASIC in a system in package (SIP).

III. MEMS-BASED PUF

MEMS sensors have an unique fingerprint based on inherent variability in silicon manufacturing processes. Since MEMS sensors are present in numerous applications, adding secure key storage capabilities would provide an additional value, making them enhanced sensors. This means there would be no need for additional devices solely for the purpose of key storage. Furthermore, considering resilience to different kinds of attacks, MEMS-PUFs offer several advantages. MEMS sensors are very complex entities with many very different features and the behavior is hard to model. Considering invasive attacks, a read-out is expected to be difficult, or in some cases even infeasible. The reason for this is that tampering with a MEMS or even with the mold package changes the properties of the MEMS and thus the key, e.g., by changing the stress conditions inserted by the packaging process or by changing the internal pressure. Hence, MEMS could provide a tamper-proof PUF without any overhead which was identified as a major future research topic in [19].

Fig. 1 shows schematically an usual example for a system in package (SIP) with a MEMS sensor and an ASIC that are encased by a mold package. MEMS and ASIC are placed on the same level, connected by wire bonds and placed on a PCB substrate with a Ball Grid Array for the electrical contacts to the environment. Alternatively, MEMS and ASIC could also be stacked vertically and connected by through-silicon vias. For high security applications, it is recommendable to carry out all security relevant operations for authentication or encryption on the ASIC. In this case, the secret key would never leave the package in order to make it infeasible for an attacker to get access to security-critical information. For this, a True Random Number Generator (TRNG) would be needed within the system additionally to the cryptographic key derived from the MEMS. One approach to derive truly random numbers could be to exploit the thermal noise as a source of entropy which is present in the measurements of the electrical capacitances between the electrodes, for example. The use of thermal noise for the generation of random numbers has already been described in previous work as in [20], [21].

On the basis of the above-mentioned assumptions, such a system would possess similar security properties as a hardware security module (HSM) [22] or a trusted platform module

(TPM) [23]. This could also be further enhanced by the development, e.g., of specific package concepts, increasing systems security. Moreover, new MEMS concepts could be exclusively designed for the use as PUFs only (dedicated MEMS-PUFs).

IV. Identification of Suitable Features

In order to identify suitable features for the use as a PUF, we have to point out initially the requirements that a feature has to fulfill. These can be derived in principle from the PUF definition.

1) *Uniqueness*. Based on the used parameters, it must be possible to identify the device absolutely uniquely. Measurable variability of the used parameters has to be inherent in the system. This variability should not be controllable even for the manufacturer in order for copying attacks to become infeasible.
2) *Robustness*. The parameters should be stable even when affected by different environmental conditions, i.e., temperature, humidity, aging.
3) *High Bit Entropy*. In case of using several parameters to derive the final response, low correlation among them should preferably exist. This is important because, the stronger the parameters correlate, the less entropy do they offer for the extracted cryptographic key.

A. Quantization Scheme

The generation of a binary key from the measured values requires a quantization procedure beforehand. The general problem of converting such analog measured values into binary strings is also known in the field of biometrics. Thus, a procedure is developed that is inspired by a method described by Chang et al. [15]. There, the authors proposed a procedure for cryptographic key generation from biometric features and verified it, as it applies to human face recognition. The modified procedure used in this work is explained below. Fig. 3 shows exemplary the quantization scheme for a Gaussian distributed parameter.

The basic factors for this procedure are the mean value μ and the standard deviation σ_{global} of the global distribution of a parameter calculated from all devices and the local variation V' which can be interpreted as the robustness of a parameter affected by temperature and measurement noise. Ideally, the cumulative distribution function for a normal distribution with our mean μ and deviation σ_{global} is given by equation (1).

The global distribution is divided into several ranges A_i with an equal probability of occurence until the whole distribution is covered with a very high probability $(6-\sigma)$. Each range has a left bound $A_{i,l}$ and a right bound $A_{i,r}$. Initially, the width of the ranges A_1 to the left and right of the global mean value μ are defined based on the value for V'. Afterwards, the further ranges $A_2, \dots A_n$ are determined so that each range occurs with the same probability, equation (2).

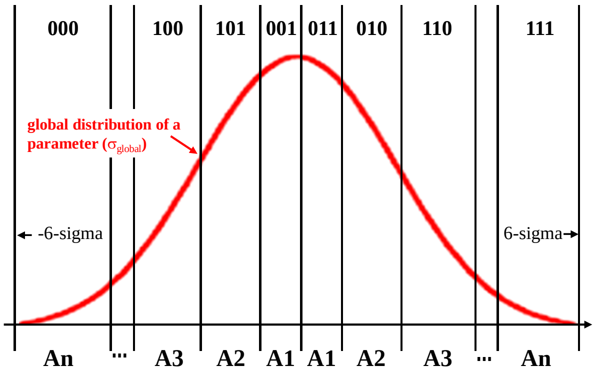


Fig. 3. Quantization scheme (exemplary for one parameter).

$$F(x) = \frac{1}{\sigma_{global}\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma_{global}}\right)^2} dt \qquad (1)$$

$$F(A_{i,r}) - F(A_{i,l}) = F(A_{i+1,r}) - F(A_{i+1,l}) \qquad (2)$$

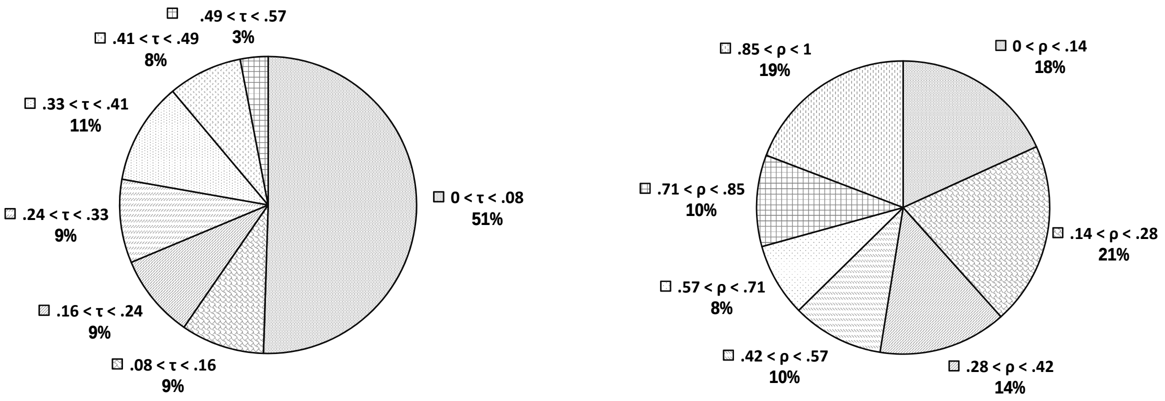
A bit combination is assigned to each range. The number of bits that can be derived from a parameter in this way can be calculated by $log_2(2 \times A_n)$. This procedure is carried out for all parameters and the key parts are concatenated to the cryptographic key seed.

B. Experimental Setup

The measurements were carried out on wafer-level, i.e., the devices are not in a mold package and laboratory measuring equipment is used for all measurements. We use the probe station PA 200 by Süss Micro Tec (Fig. 4) which enables to measure a large number of devices fully automated and the setting of temperature by a heatable chuck. Furthermore, the test equipment consists of a multiplexer probe card for driving and measuring on the different electordes and the Impedance Analyzer 4294A by Agilent Technologies. For contacting the sensor pads, a device with several contact probes is mounted on the probe card.

The device under investigation was a 3-axis gyroscope. We measured all parameters that are mentioned above (Section II) for each channel so that we had in total more than 50 parameters of 70 devices. We repeated the measurements multiple times at room temperature (RT) to determine the repeatability of the measurements. Additionally, we carried out the measurements at 85 °C to verify the robustness of the parameters at higher temperature. As a result of the repeated measurements and the temperature variation, we can describe the parameter robustness as combination of a Gaussian distributed factor f_{noise} which is based on measurement noise and a temperature dependent shift factor f_{shift}. Thus, the local variation V' of a parameter can be estimated from a measured value V and this two factors in the following way:

$$V'(T) = f_{noise}V + f_{shift}(T) \qquad (3)$$



(a) Percentage distribution of ratios $\tau = \frac{V'_{max}}{\sigma_{global}}$.

(b) Percentage distribution of correlation coeffients ρ between the used parameters.

Fig. 2. Percentage distribution of ratios and correlation coeffients.

Hence, the maximum local variation V'_{max} occurs in case of the maximum temperature range (from RT to $85\,°\mathrm{C}$) and an adding effect of the factors f_{noise} and f_{shift}.

Initially, we identify basic suitable parameters regarding the ratio τ of the maximum local variation V'_{max} to the global variation σ_{global} for each parameter $\frac{V'_{max}}{\sigma_{global}}$. The ratio $\tau = \frac{V'_{max}}{\sigma_{global}}$ should be significantly smaller than 1.

C. Parameter Results

As mentioned above, major influence factors on the parameter variation are the variation of the geometric dimensions (structure width and thickness). For this reason, some of the parameters are strongly correlated with this factors. Because all measurement variables depend on them in a similar way, an appropriate measure to reduce this dependency is to calculate ratios. Thus, other effects become more important such as local differences in the structure widths, for example.

Regarding the frequency modes, the use of ratios provides a further advantage. The frequency modes are temperature dependent in a linear way because of the temperature dependence of the Young's modulus [24]. Thus, the frequency modes themselves vary about temperature with a constant factor. This factor is deleted by calculating ratios and, hence, the ratios are significantly more stable about temperature than the pure frequency modes. This also applies for the capacitances in a similar manner.

As a first result of the measurements, we can define the following parameters as potentially appropriate (in brackets is the quantity of a parameter type):

- frequency modes (9),
- capacitances (6),
- quadrature signals (2).

Hence, the further evaluations are based on this parameters. Fig. 2a shows the percentage distribution of the τ values for this properties. The proportion of 50% of ratios τ between 0 and 0.08 is mainly originated from the frequency mode based parameters and the quadrature signals. The higher τ values come from the ratios of capacitances. This is mainly caused by their low σ_{global}-values.

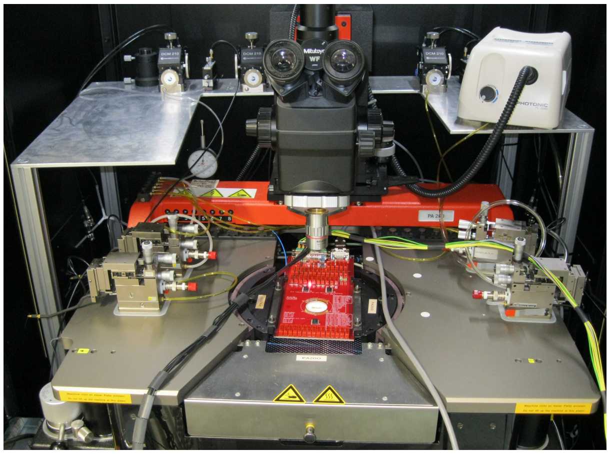


Fig. 4. Probe station PA 200 by Süss Micro Tec with mounted probe card used for measurements on wafer-level.

In terms of cryptographic key generation, the consideration of the correlation between the parameters is of fundamental importance. We determine the correlations between all suitable parameters. The correlation coefficient $R_{X,Y}$ between two parameters X and Y with N measurement values is calculated by equation (4), whereas $C = \begin{pmatrix} Cov(X,X) & Cov(X,Y) \\ Cov(Y,X) & Cov(Y,Y) \end{pmatrix}$ is the covariance matrix. The covariance $Cov(X,Y)$ of X and Y is given by equation (5).

$$R_{X,Y} = \frac{C_{X,Y}}{\sqrt{C_{X,X}C_{Y,Y}}} \tag{4}$$

$$Cov(X,Y) = \frac{1}{N-1}\sum_{i=1}^{N}(X_i - \mu_X)(Y_i - \mu_Y) \tag{5}$$

Fig. 2b shows the percentage distribution of correlation coefficients ρ between the used parameters.

As mentioned above, the stronger the parameters correlate, the less entropy do they add to the key. For this reason we have to define an upper limit for the correlation coefficients ρ_{max} that we accept. Parameters that are stronger correlated than this upper limit were rejected. The choice of this limit affects obviously the number of bits that can be derived. Table I shows the dependence of the number of bits on ρ_{max}. To analyze the effect of the upper limit value, we vary them stepwise and estimate the entropy of the extracted keys by different methods (see Section VI).

TABLE I
DEPENDENCE OF THE NUMBER OF DERIVABLE BITS ON THE CORRELATION UPPER LIMIT ρ_{max}.

ρ_{max}	.50	.62	.74	.86	.98
bits	30	30	38	63	138

V. SIMULATING PUF RESPONSES

In order to generate an arbitrarily number of keys we make Monte-Carlo simulations. Based on this, we are able to generate keys from both different devices and many keys from a single device.

A. PUF Responses from Different Devices

The simulation of PUF responses from different devices allows us to test if the results of the entropy estimation are affected from the limited length of our measured bit streams. For the simulation we can assume that all of the parameters are Gaussian distributed. Then, we have to consider the mean value μ and the standard deviation σ_{global} of the global distribution of the parameters and the correlation matrix R that contains all correlation coefficients. The procedure is as follows:

1) generation of a normally distributed random number matrix Z with dimensions (number of keys i, number of parameters j)
2) Cholesky decomposition of the correlation matrix R which is based on the measurements $R = GG^T$
3) multiplying Z with G to receive the normally distributed random number matrix Z_R considering the correlations of R $Z_R = ZG$
4) generation of matrix $P_{MC}(i,j)$ with parameter values $P_{MC}(i,j) = \mu(j) + \sigma_{global}(j)Z_R(i,j)$

B. Maximal Bit Error Rate Estimation

The estimation of a maximal Bit Error Rate (BRR_{max}) is of great significance. The BRR denotes the difference between two keys of the same device generated at different times or environmental conditions (e.g., different temperatures) and it is also known as the intra-distance which is a measure for the robustness of a key. The BRR should be preferably 0, however, due to the noisy nature of physical measurements, this is not always achieved in practice.

TABLE II
BRR_{max} FOR DIFFERENT VALUES OF ρ_{max} WITH THE ASSOCIATED PROBABILITIES $P(BRR > BRR_{max})$ FOR A BRR ABOVE BRR_{max}.

ρ_{max}	BRR_{max}	P	BRR_{max}	P	BRR_{max}	P
.50	9	3.19e-6	10	4.18e-7	11	5.02e-8
.62	9	1.26e-6	10	1.48e-7	11	1.61e-8
.74	10	9.05e-7	11	1.18e-7	12	1.42e-8
.86	11	8.83e-7	12	1.29e-7	13	1.74e-8
.98	19	3.44e-6	20	9.39e-7	21	2.45e-7

Because of PUF variability across different environmental conditions and measuring inaccuracy, when a PUF is challenged a noisy response is obtained. In applications where the PUF response is used as a cryptographic key a noisy response is not acceptable. To solve this problem, algorithms known as fuzzy extractors leverage non-secret helper data to work around the noisy nature of physical measurements typical of PUF applications (see Section VII). However, such a bit error correction result in an entropy loss and means a reduced key length. The amount of reduction depends on the number bit-flips that have to be corrected. This has to be assessed by the BRR_{max} estimation.

In order to be able to estimate the robustness of a parameter, we repeated our measurements multiple times. As we can describe the variability by Equation 3, we carry out a Monte-Carlo simulation to determine the probabilities for dedicated bit error rates. Therefore, we create a normally distributed random number matrix Z with dimensions (number of keys i, number of parameters j) to receive the local variation of the parameters for a device $V'(i,j)(T) = f_{noise}Z(j)V(i,j) + f_{shift}(j)(T)$.

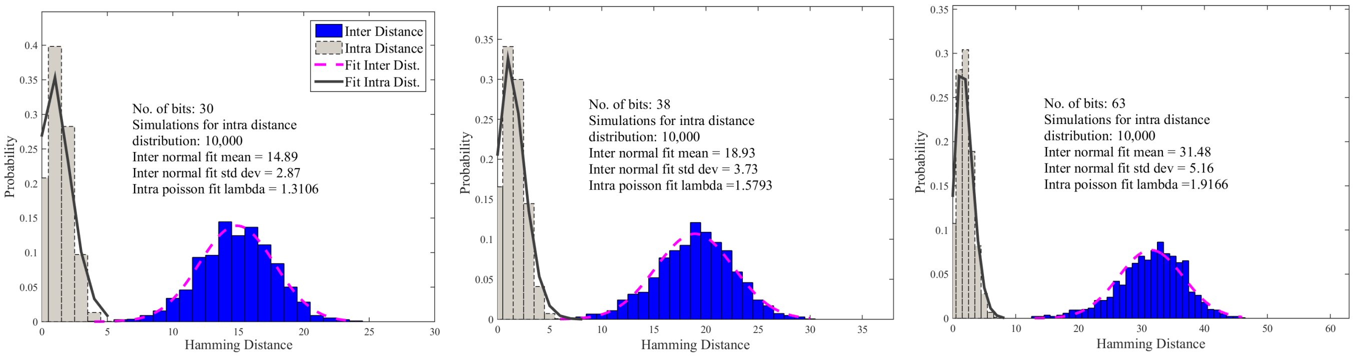
We estimate the BRR_{max} for different values of ρ_{max} with the associated probabilities $P(BRR > BRR_{max})$ for a BRR above BRR_{max}. The probabilities are calculated from a Poisson distribution fit (see Fig. 5). The results are presented in Table II. The values of each row are based on 10,000 keys which are created by the Monte-Carlo simulation.

VI. ENTROPY ESTIMATION

An important aspect PUFs should show, besides robustness, is randomness. This means that given all responses from all PUF devices, an attacker should have a negligible chance of estimating a future response of a PUF. Also the bits in a response should be random and unpredictable, so that chances for two responses from two different PUFs to be "close" are negligible small.

In order to assess the randomness of our PUF design, we use the following methods:

1) Inter and Intra Hamming Distances: To evaluate the potential of physical properties for PUF applications, the ability to uniquely identify each instance is essential. This can be formally defined by the concept of inter and intra Hamming distances. The inter distance HD_{inter} depicts the difference between two keys of different devices and it is a measure for key uniqueness. The Intra-Distance HD_{intra} denotes the difference between two keys of the same device generated at different times or environmental conditions (e.g., different temperatures). The Intra-Distance is a measure for the robustness of a key and determines directly the number



(a) ρ_{max} = .62.

(b) ρ_{max} = .74.

(c) ρ_{max} = .86.

Fig. 5. Inter and intra Hamming distance distributions of measured data.

of bit-flips. An ideal PUF yields a $HD_{intra} = 0\%$ and $HD_{inter} = 50\%$.

2) CTW Compression: We try to compress our responses with CTW (Context Tree Weighting), a lossless compression algorithm [25]–[27]. This method is optimal for stationary ergodic sources and gives an optimal compression. The resulting compression on bit strings often used to estimate the entropy rate [28]. The idea is that bit sequences with full entropy cannot be compressed, meaning if a lossless compression is possible, then our responses do not have full entropy. Thus, CTW gives an upper bound on entropy.

3) NIST Randomness Test: We use the PUF responses as input to the NIST randomness test suite [29] to verify, if enough of these tests pass. This would indicate full entropy with high probability. We configured each test in NIST SP800-22 in the same manner as in [30], meaning the significance level of each test is set to 1%, so that 99% of the test samples pass if the input was truly random. Let the number of samples be n and the probability of passing each test is p, then the number of passing samples follow a binomial distribution. The value p' of observed passings is then defined as

$$p' = p \pm 3\sqrt{\frac{p(1-p)}{n}} \qquad (6)$$

Also the NIST tests yield a P-value, generated by a χ^2 test, which indicates randomness on a an uniformly distributed assumption if the P-value is ≥ 0.0001. In order to pass a NIST test both conditions must be fulfilled – the proportion of passed tests should exceed the above the threshold defined above and the P-value should be above 0.0001.

4) NIST Min-Entropy Estimation: Since CTW only gives us an upper bound on entropy and the NIST randomness test suite yield test results for full entropy or not, we try to estimate the min-entropy with tests mentioned in NIST's special publication 800-90B [31], indicating a lower bound of entropy for our purposes.

Our source is not independent and identically distributed (non-IID), because we have seen so far in the previous sections that there are correlations in the bit strings. So, we tested our PUF responses with the following five estimations for non-IID sources [31]. Each test yields an estimation on min-entropy and the overall estimated min-entropy is the minimum of these five values. The tests are configured with a confidence level of 95%.

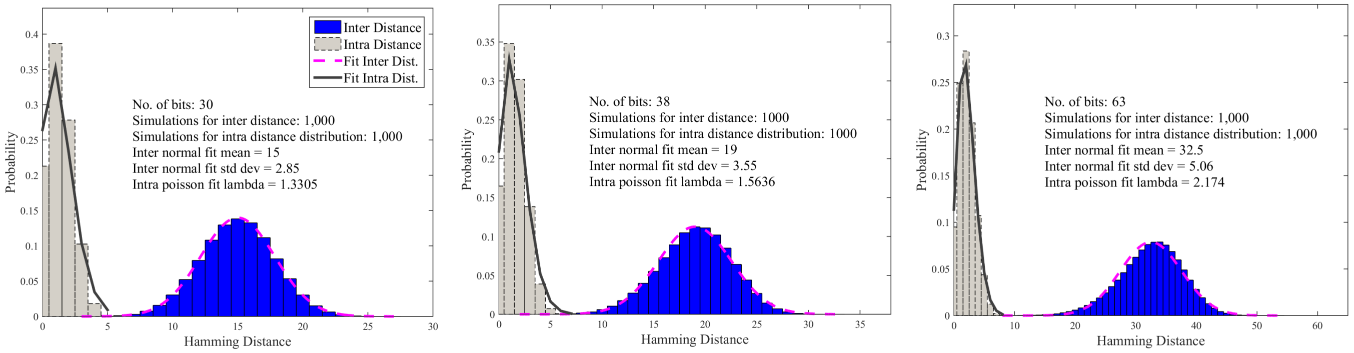
a) Collision Test: The collision test measures the mean time to the first collision in a dataset. Based on these collision times, the collision statistic tries to estimate the probability of the most-likely state. For biased noise sources toward an output or state the test will result in a low entropy estimate, say when there is a short mean time until a collision. Longer mean times on collisions end up with in higher entropy estimates.

b) Partial Collection Test: The partial collection test computes the entropy of a dataset based on how many distinct values in the output space are observed. Low entropy estimates are output for datasets that contain a small number of distinct symbols, and high entropy estimates are the output when the bit strings diversify quickly.

c) Markov Test: The Markov test consists of different Markov processes, from first-order up to n^{th}-order. In a first-order Markov process, the output state depends only on the current state and in an n^{th}-order Markov process, the output state depends on the current and all previous n-1 states. To detect dependencies, the test builds a Markov model to be used as a template for a given source. The min-entropy estimates result from measuring the dependencies between consecutive outputs from the noise source. Thereby the estimates are not based on an estimate of min-entropy per output, but on the entropy present in any chain of outputs.

d) Compression Test: The compression test estimates the entropy rate by compressing the input data set. As compression method the Maurer Universal Statistic [32] is used. It generates a dictionary of values, and then computes the average number of samples required to write an output based on the dictionary.

e) Frequency Test: The frequency statistic models the probability distribution of the given data set. The entropy estimation is based on the occurrence of the most-likely symbol.



(a) ρ_{max} = .62.

(b) ρ_{max} = .74.

(c) ρ_{max} = .86.

Fig. 6. Inter and intra Hamming distance distributions of simulated data.

TABLE III

CTW COMPRESSION RATES ON REAL DEVICE MEASUREMENTS FOR DIFFERENT UPPER CORRELATION LIMITS ρ_{max}. NOTE THAT THE DATA SHOWS AN UNCOMPRESSABILITY, DUE TO THEIR SMALL SIZE AND IS MENTIONED FOR THE PURPOSE OF VERIFICATION.

ρ_{max}	Size uncompressed (bytes)	Size compressed (bytes)	compression rate of measurements (bits/byte)	compression rate random file (bits/byte)
.50	148	165	8.25676	8.23649
.53	164	181	8.22561	8.18902
.56	164	181	8.22561	8.18902
.59	164	181	8.22561	8.18902
.62	164	181	8.22561	8.18902
.65	192	209	8.20312	8.18229
.68	254	272	8.16929	8.14173
.71	254	272	8.16929	8.14173
.74	295	313	8.15254	8.13559
.77	331	349	8.12991	8.12085
.80	292	309	8.13356	8.13356
.83	413	432	8.12107	8.10412
.86	451	470	8.11973	8.10200
.89	496	515	8.10282	8.09476
.92	605	624	8.0843	8.08099
.95	645	664	8.08062	8.07752
.98	978	998	8.05828	8.05419

A. Entropy Estimation of Measured Data

We estimated the entropy of the responses with several different upper correlation limits ρ_{max} from the 70 measured devices.

1) Inter and Intra Hamming Distances: Fig. 5 shows the inter and intra Hamming distance distributions of the measured data for three different values of ρ_{max}. The inter distance distribution is fitted by a normal distribution. The mean of the fit is close to 50%. The intra distance distribution is based on the Monte Carlo simulation (10,000 runs) that we explained in Section V-B. To be able to identify a device securely, it is important that the intra and intra distance distributions overlap just with negligible probability, which is the case here. The best result do we receive for ρ_{max} = .86.

2) CTW Compression: The compression method was configured with a tree depth of 6 and we used a Krichevski-Trofimov estimator [25]. It is important to note, that CTW compression does not work efficiently with the small sizes we give here as input, so all resulting compression rates are above 100%. Still, would the bit strings have major statistical defects, then a compression would be possible even with these small input sizes. For the purpose of verification we also tried to compress truly random bits with the same input sizes as our responses, yielding similar results. Therefore, our bit strings show an uncompressability. The results can be found in Table III.

3) NIST Randomness Test: We used the NIST randomness tests as described in Section VI-3 on our bit strings. The minimum pass p' rate for each statistical test is approximately 8, because we chose our number of samples $n = 10$. The results consist of two values per test and one symbol – the first value is the P-value and the second value represents the number of passed runs p, where $p \geq p'$ to pass a test. The third symbol indicates if all conditions for a passed test are met (✓) or not (×). The results indicate a high entropy in our bit strings, since all except three tests fail. Nevertheless, the tests are not that meaningful because the input size to these tests is very small.

4) NIST Min-Entropy Estimation: Due to the short overall bit strings we derived from our measurements, the NIST Min-Entropy Estimation were not able to calculate valid results. So we omit these tests in this section.

B. Entropy Estimation on Simulated PUF Responses

We estimated the entropy of bit strings, which offspring from our measurements from real devices. However, the generated bit strings are not long enough to generate meaningful results on entropy estimation. Therefore we repeat the entropy estimation on simulated data, too. For a conservative estimate we choose the minimum of our estimated entropy value for further constructions.

We also validated to concatenate and partly replace simulated bits with the one from our real measurements and we found no significant difference.

1) Inter and Intra Hamming Distances: Fig. 6 shows the inter and intra Hamming distance distributions of the simulated data (1,000 runs for both intra and inter distances) for the same

TABLE IV

NIST RANDOMNESS TESTS FOR DIFFERENT UPPER CORRELATION LIMITS ρ_{max}. THE FIRST VALUE PER TEST IS THE P-VALUE. THE SECOND VALUE IS THE NUMBER OF PASSED RUNS p, WHERE $p \geq p'$ TO PASS A TEST. THE THIRD SYMBOL S INDICATES IF ALL CONDITIONS FOR A PASSED TEST ARE MET (✓) OR NOT (×).

ρ_{max}	Frequency			Block Frequency			Cumul. Sums			Runs			FFT			Approx. Entropy			Serial			Linear Complexity		
	P-value	p	S	P-value	p	S	P-value	p	S	P-value	p	S	P-value	p	S	P-value	p	S	P-value	p	S	P-value	p	S
.50	.03517	9	✓	.01791	9	✓	.54520	8.5	✓	.35049	10	✓	.00430	10	✓	.53415	10	✓	.23641	10	✓	.00009	10	×
.53	.03517	9	✓	.12233	9	✓	.06688	9	✓	.91141	10	✓	.00430	10	✓	.03517	10	✓	.37373	9.5	✓	.35049	8	✓
.56	.03517	9	✓	.12233	9	✓	.06688	9	✓	.91141	10	✓	.00430	10	✓	.03517	10	✓	.37373	9.5	✓	.35049	8	✓
.59	.03517	9	✓	.12233	9	✓	.06688	9	✓	.91141	10	✓	.00430	10	✓	.03517	10	✓	.37373	9.5	✓	.35049	8	✓
.62	.03517	9	✓	.12233	9	✓	.06688	9	✓	.91141	10	✓	.00430	10	✓	.03517	10	✓	.37373	9.5	✓	.35049	8	✓
.65	.53415	8	✓	.06688	9	✓	.32824	8	✓	.73992	10	✓	.00095	10	✓	.73992	10	✓	.44232	10	✓	.00204	8	✓
.68	.73992	10	✓	.53415	9	✓	.43112	9.5	✓	.12233	10	✓	.00888	10	✓	.06688	10	✓	.63703	10	✓	.35049	10	✓
.71	.73992	10	✓	.53415	9	✓	.43112	9.5	✓	.12233	10	✓	.00888	10	✓	.06688	10	✓	.63703	10	✓	.35049	10	✓
.74	.91141	10	✓	.53415	9	✓	.54520	9.5	✓	.73992	9	✓	.00095	10	✓	.00888	10	✓	.63095	9	✓	.21331	9	✓
.77	.21331	9	✓	.53415	10	✓	.44232	9.5	✓	.01791	10	✓	.00888	10	✓	.53415	10	✓	.63703	10	✓	.01791	10	✓
.80	.35049	10	✓	.73992	10	✓	.53415	10	✓	.91141	10	✓	.06688	9	✓	.35049	9	✓	.63703	9.5	✓	.12233	10	✓
.83	.99147	10	✓	.53415	10	✓	.63703	9.5	✓	.35049	10	✓	.00888	9	✓	.21331	10	✓	.53415	10	✓	.73992	10	✓
.86	.21331	10	✓	.21331	10	✓	.44232	10	✓	.21331	10	✓	.03517	10	✓	.21331	10	✓	.40340	9.5	✓	.35049	10	✓
.89	.35049	10	✓	.53415	10	✓	.14010	10	✓	.53415	10	✓	.00204	10	✓	.53415	10	✓	.53415	10	✓	.12233	10	✓
.92	.03517	9	✓	.53415	10	✓	.23641	8	✓	.35049	10	✓	.01791	10	✓	.53415	10	✓	.63703	9.5	✓	.91141	10	✓
.95	.73992	9	✓	.00430	8	✓	.14010	9	✓	.73992	10	✓	.35049	10	✓	.91141	10	✓	.51687	10	✓	.06688	10	✓
.98	.73992	8	✓	0	5	×	.00107	7	×	.91141	9	✓	.01791	9	✓	.73992	9	✓	.44232	9.5	✓	.53415	10	✓

values of ρ_{max}. The results are comparable to those from the measured data.

2) CTW Compression: Again, we configured the compression method with a tree depth of 6 and we used a Krichevski-Trofimov estimator [25]. The compression rate is given in bits per byte, meaning that bit strings with full entropy result in a compression rate of 8 bits/byte. Our compression results indicate, that the quantized bit strings up a correlation upper limit ρ_{max} of 0.71 have nearly full entropy. With an increasing ρ_{max} the compression rate drops. The results can be found in Table VIa. Since CTW compression gives us an upper bound on the entropy, meaning the entropy of our bit strings can be less, but not more, this bound is also given in Fig. 7.

3) NIST Randomness Test: We used the NIST randomness tests as described in Section VI-3 on our simulated bit strings. The minimum pass rate p' for each statistical test is approximately 96, because we chose our number of samples $n = 100$. However, most of the NIST randomness tests failed, so we omit the actual results at this place. We hypothesize the reasons are that our bit strings do not have full entropy, but nearly full entropy as seen in Table VIa, and that the random number generator used for generating the simulated bit strings is not truly random itself.

4) NIST Min-Entropy Estimation: The five tests for a min-entropy estimation were configured to analyze 8-bit symbols, to have a comparable symbol size as the CTW compression. Four tests gave invalid results, indicated with a $\perp$, as output. We also verified the estimated min-entropy values with a symbol size of 16 bits, where all results were valid, and the estimations were similar to the 8-bit symbol tests. However, our results show that the Markov test always produces the lowest min-entropy estimate, so the other tests do not come into account anyway. The results can be found in Table VIb.

As the results for an estimated min-entropy give us an estimated lower bound on entropy of our bit strings. Fig. 7 shows the upper and lower bounds on entropy depending on the chosen upper correlation limit ρ_{max} as a combined result of CTW compression and min-entropy estimation.

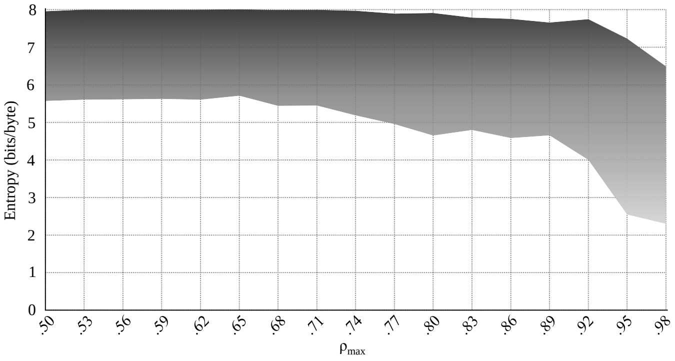


Fig. 7. Entropy upper and lower bounds as function of correlation coefficient.

VII. KEY ALIGNMENT

Fuzzy Extractors [17] can be used to extract the same cryptographic keys from correlated measurements, i.e. noisy PUF measurements. The keys are generated in an enrollment phase and, when the PUFs are in the field, can be reconstructed with a previously generated helper data P. This helper information should leak no information of the key, whatsoever, so it can be stored in an external memory on the PUF device itself or can be transmitted over the internet. Our construction can be easily adapted to be secure against an active attacker on the helper data. With a robust fuzzy extractor [33] we would introduce a message authentication code (MAC), which can be used to authenticate the helper data.

The correctness property of fuzzy extractors state that the construction outputs the exact same key if the distance between

TABLE V
CTW COMPRESSION AND MIN-ENTROPY ESTIMATION RESULTS FOR SIMULATED BIT STRINGS.

ρ_{max}	Compression rate
	(bits/byte)
.50	7.95728
.53	7.99916
.56	7.99914
.59	7.99943
.62	7.99953
.65	8.00645
.68	7.99308
.71	7.99439
.74	7.97247
.77	7.89551
.80	7.91321
.83	7.78784
.86	7.75614
.89	7.65709
.92	7.74842
.95	7.23375
.98	6.49315

(a) CTW compression rates of simulated PUF responses for different upper correlation limits ρ_{max}.

ρ_{max}	Collision	Partial collection	Markov	Compression	Frequency	Estimated
	test	collection	test	test	test	min-entropy
.50	6.22521	5.90069	5.56694	6.02559	7.11177	5.56694
.53	6.82882	6.64823	5.60197	6.69622	7.31813	5.60197
.56	6.80155	6.63535	5.60936	6.65570	7.32902	5.60936
.59	6.66015	6.64927	5.62541	6.61198	7.31514	5.62541
.62	6.68146	6.53839	5.60263	6.58637	7.32317	5.60263
.65	6.91785	6.92833	5.70984	6.88106	7.46459	5.70984
.68	$\perp$	7.18993	5.43725	$\perp$	7.69332	5.43725
.71	$\perp$	7.14051	5.44469	$\perp$	7.67901	5.44469
.74	6.23016	6.21762	5.18516	6.22934	6.92685	5.18516
.77	6.59699	5.80079	4.95366	6.07370	6.76764	4.95366
.80	6.95015	5.78587	4.65017	6.07816	7.09745	4.65017
.83	7.38529	5.67176	4.79813	6.03277	7.19481	4.79813
.86	5.80063	5.39787	4.57970	5.61252	7.13658	4.57970
.89	5.78053	5.06731	4.65375	5.41692	6.97970	4.65375
.92	5.72708	5.07137	3.99810	5.38990	6.53882	3.99810
.95	4.20345	3.64993	2.54841	4.05859	5.62399	2.54841
.98	3.70251	2.92305	2.29599	3.42860	5.04268	2.29599

(b) NIST tests for min-entropy estimation. The estimated min-entropy is in bits per byte. Note that tests yielding an invalid result, output a $\perp$.

two measurements w and w' is smaller than some error T, denoted as $dis(w, w') \leq t$.

A. Error Correction

We choose the syndrome construction from [17] to reconcile our measurements w and w' and followed the idea of [34] to get parameters for our setting, because recent research shows that an alternative, i.e. two-stage concatenated codes with repetition codes, can be very risky [35]. For, e.g., the setting with $\rho_{max} = 0.86$ we use a $[n = 63, k = 10, t = 13]$-BCH code, capable of correcting 13 errors in a 63-bit code word. The entropy loss of this construction to an eavesdropper is $n - k = 53$ bits. The extracted message has 10 bits after error correction. We optimized the quantization process, so that the resulting response w has at most $t = 13$ errors with a probability of $1.74 \cdot 10^{-8}$, as given in Table II. For a cryptographic 128-bit key, we need to combine the min-entropy results from Table VIb and the chosen code, so that we need

$$\left\lceil \frac{\textit{length key}/\textit{min-entropy rate}}{\textit{length message}} \right\rceil = \left\lceil \frac{128/0.5725}{10} \right\rceil = 23$$

PUF responses. This means the overall PUF response, concatenated from 23 sensors, has a length of $23 \cdot 63 = 1149$ bits and that our overall helper data P has a length of $23 \cdot 53 = 1219$ bits. Putting it all together, we receive an overall authentication failure, due to decoding failure, with a probability of $1 - (1 - 1.74 \cdot 10^{-8})^{23} = 4.00 \cdot 10^{-7}$. This is less than the required standard of at most one failure per one million uses. Note that, despite our responses do not have full entropy, our parameters are an improvement of needed response and helper data bits, compared to [34] while having roughly the same false rejection rate.

B. Randomness Extraction

To generate a strong secret key, we finally hash our corrected codeword. The lightweight hash function SPONGENT [36] seems to be a perfect candidate for a resource-constrained sensor device. In particular, we chose the SPONGENT-128/256/128 construction, which has an 128-bit output with full preimage and second-preimage security. To carry on with the previous example, we hash the corrected 1449-bit code word with a min-entropy rate of 0.5725 to receive a 128-bit key with full entropy.

VIII. DEDICATED MEMS-PUF DESIGN

We showed that there are several sensors necessary to derive a 128-bit key based on our used parameters. This could be possible in applications in which several sensors are existent (e.g., 9-degrees-of-freedom sensor node). Another option is to design a specific MEMS element for security purposes only. Such a dedicated MEMS-based PUF could be realized in an area saving manner and it can be optimized providing at least the same number of suitable properties for the use as PUFs as an usual gyroscope. Furthermore, the structures of such a specific MEMS could be designed in a way that increase the variability of the properties to derive more bits from a single parameter. One example is the use of the minimum beam width for the springs in order to increase the percentage influence of the beam width variation. The aim of increasing variability could be achieved by measures in the manufacturing process as well because this is optimized actually to keep variations at a minimum.

Fig. 8 illustrates our proposal for a dedicated MEMS-based PUF concept. It is a 3-masses oscillator that is free to move in all spatial dimensions. The masses are linked by doubling U-springs which are very sensitive to asymmetries that should

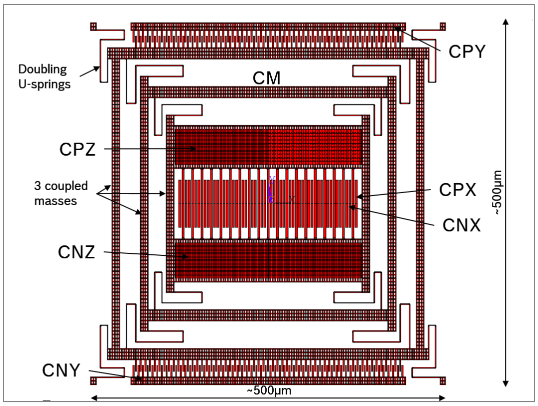


Fig. 8. Dedicated MEMS PUF design.

increase the quadrature signals and the whole structure is suspended by four doubling U-springs at the outside corners. The system can be driven and measured by the electrode pairs CPX/CNX, CPY/CNY in case of in-plane movements and CPZ/CNZ in case of out-of-plane movements with respect to the potential of the masses (CM).

The structure contains twelve frequency modes which are illustrated in Fig. 9. Three frequency modes are based on in-plane movements in y direction (9a, 9e, 9g) and three ones in x direction (9b, 9d, 9i). Furthermore, there are six frequency modes for out-of-plane movements. Three ones for translational motions (9c, 9h, 9k) and three ones for rotational motions (9f, 9j, 9l). We are able to drive and measure all of these mode shapes. A big advantage of such a dedicated MEMS-based PUF is that it is possible to design the mechanical structure in a way that the usable frequency modes are close together and optimally defined for the use as PUFs. This is in contrast to the structure of a MEMS-based gyroscope where the focus is on the drive and detection modes shifting all further frequency modes as far as possible away from them. Additionally, there is a quadrature signal for each frequency mode and six pairs of electrodes, i.e., the design provides in total twelve frequency modes, twelve quadrature signals and six electrical capacitances.

To estimate the number of bits that could be derived from our structure, we carry out FEM-simulations using ANSYS to calculate the frequency modes. Subsequently, we determine the capacitances between the electrodes and the quadrature signals with a reduced order model developed by Gugel [37] which is based on the principle of modal superposition. This method transmits the equation of motion (Equation 7) used in the FEM-analysis to a description of the system with reduced complexity solving the eigenvalue problem (-ω_i^2M + K)φ_i = 0 with the eigenvectors φ_i and the eigenvalues ω_i. As a result, we receive the transformation matrix Φ including the eigenvectors φ_i. M is the mass matrix, K is the stiffness matrix and D is the damping matrix. Equation 10 describes the system in the modal space with the deflections q whereby x = Φq.

TABLE VI
Number of derivable bits depending on the correlation upper limit ρ_{max}.

ρ_{max}	.50	.62	.74	.86	.98
bits	62	73	89	110	199

$$M\ddot{x} + D\dot{x} + Kx = F \tag{7}$$

$$M\Phi\ddot{q} + D\Phi\dot{q} + K\Phi q = F \tag{8}$$

$$\Phi^T M\Phi\ddot{q} + \Phi^T D\Phi\dot{q} + \Phi^T K\Phi q = \Phi^T F \tag{9}$$

$$\tilde{M}\ddot{q} + \tilde{D}\dot{q} + \tilde{K}q = \tilde{F} \tag{10}$$

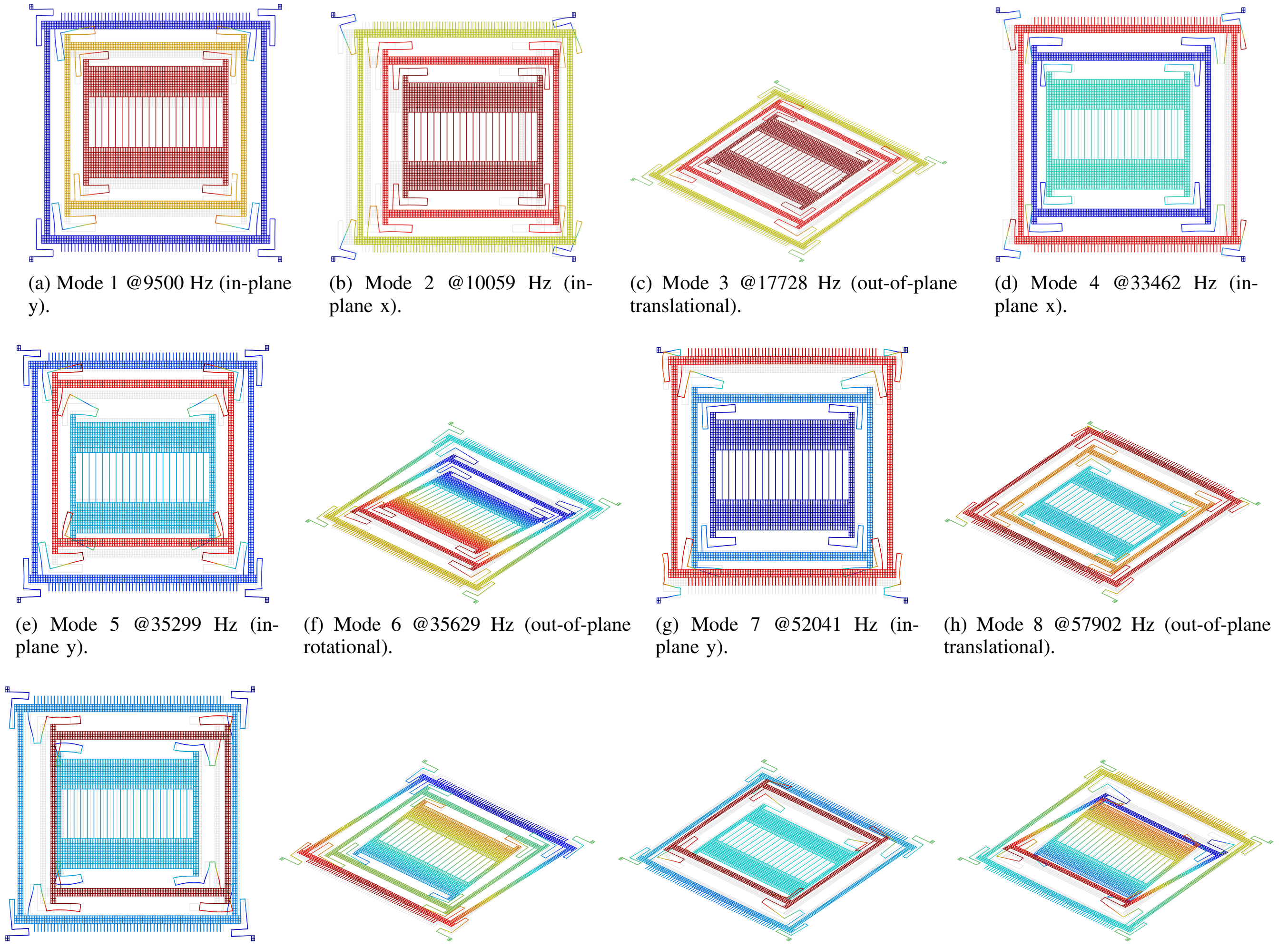
For simulations, we consider the following aspects of manufacturing process-related variations:

- geometric dimensions (structure width and thickness),
- slight differences of the beam widths locally on the legs of the U-springs,
- pressure inside the cavity,
- differences in side wall inclination.

We make 1,000 simulations of the design to estimate the key length that can be derived from the structure depending on the correlation upper limit. For the key generation procedure, we assume the same measurement accuracies and temperature dependencies as determined by the measurements of gyroscopes before. Table VI shows that it is presumably possibly to derive more bits than from the investigated gyroscopes. Note that we consider for this simulations measures in the design only. A further lengthening of the key can be easily achieved by ”worsen” the manufacturing process. Furthermore, due to the small dimensions of the structure it is conceivable to combine several of this structures in one unit concatenating their keys or to add such a structure to existing MEMS sensors for key storage purposes.

IX. Conclusion

MEMS sensors exhibit great potential for the generation of cryptographic keys. In this work, we show that MEMS-based gyroscopes, which have been developed for a broad range of capabilities, can be used to derive a high entropy cryptographic key. We identify properties of MEMS-based gyroscopes, suitable for PUF applications by a large number of measurements on wafer-level. In order to quantize the measurement values, we propose for an appropriate procedure. We verify the uniqueness and reliability of the generated bit strings. Furthermore, we estimate upper and lower bounds on the entropy of these bit strings and show how to implement a fuzzy extractor to derive a full entropy key from the most conservative entropy estimations. Based on error correction and randomness extraction we display the number of required devices for a 128-bit key generation from MEMS-based gyroscopes. Additionally, we present a dedicated MEMS PUF design, solely for usage as a primitive in security applications. This design is optimized in terms of potential features and chip area, allowing us to derive a full entropy 128-bit key from just



(a) Mode 1 @9500 Hz (in-plane y).

(b) Mode 2 @10059 Hz (in-plane x).

(c) Mode 3 @17728 Hz (out-of-plane translational).

(d) Mode 4 @33462 Hz (in-plane x).

(e) Mode 5 @35299 Hz (in-plane y).

(f) Mode 6 @35629 Hz (out-of-plane rotational).

(g) Mode 7 @52041 Hz (in-plane y).

(h) Mode 8 @57902 Hz (out-of-plane translational).

(i) Mode 9 @63992 Hz (in-plane x).

(j) Mode 10 @71521 Hz (out-of-plane rotational).

(k) Mode 11 @79420 Hz (out-of-plane translational).

(l) Mode 12 @96497 Hz (out-of-plane rotational).

Fig. 9. 12 frequency modes of our design proposal for a dedicated MEMS-based PUF.

a few of such structures, while still being able to fit in a single unit.

A. Limitations and Further Research

We showed that deriving a cryptographic key from a MEMS is feasible. However, we are still in need to extract more bits from the MEMS structure itself, enhancing following steps in the key generation process. Regarding the implementation of MEMS-based PUFs in sensor systems and the achievement of a further key lengthening, two approaches are possible.

1) Use of several existent MEMS sensors in a sensor system, e.g., 9 degree-of-freedom sensor nodes, and add-up of cryptographic key seeds which can be derived from the individual sensors.
2) Development of a specific MEMS-based PUF device, optimized for PUF applications.

The first approach provides an additional value for existing sensors and aims at its enhancement. This requires further investigations of MEMS-based sensors. On one hand, there could be more suitable parameters than that we have actually measured. For example in case of gyroscopes, there should be more frequency modes existent than nine. Additionally, it is possible to measure a quadrature signal for each frequency mode but we measured just two, because of constraints on the measurement setup. Especially, the quadrature signals are potentially able to lengthen the derivable keys, because they can be used to extract proportionally many bits and show little correlation with other parameters. On the other hand, investigations of different MEMS sensors have to be done. Besides, further tests should be carried out to analyze the reliability of different parameters. For example, these could be tests on packaged devices as mechanical stress tests and aging tests.

The second approach aims at the development of a dedicated MEMS-PUF which benefits from the experiences gained from investigations on different existing MEMS sensors. The design and the manufacturing process can be optimized to increase variability and thus deriving more bits per parameter. Moreover, such a specific design can be optimized that it provides more suitable parameters for PUF applications than a standard MEMS sensor. Therefore a dedicated MEMS PUF would present an excellent candidate for high security applications. Due to the small size of such an element, it is also conceivable to add this structure to a MEMS sensor without making them significantly larger or affecting its functionality.

Besides the construction of an actual PUF, estimation on min-entropy is an open research direction. State-of-the-art estimations, e.g., CTW compression, focus on giving an upper bound of entropy, leaving the problem of possible less entropy open. Clearly, for high security applications a sound estimate of the enclosed lower entropy bound should be given.

B. Related Work

1) Physical Unclonable Functions: PUFs have been divided into two categories depending on the number of uncorrelated CRPs that they accept. These two categories are strong PUFs and weak PUFs (also called obfuscating PUFs [38]), originally introduced in [9] and further developed in [38], [39]. Rührmaier et al. has formalized the strong PUF definition [38]. Their model postulate that an attacker has access to an oracle, which replies to a challenge C_i with the same response R_i as the real system. Thus, concepts that protect the access to the PUF are not taken into account, although they would lead to increased security. Examples include concepts such as controlled PUFs which protect the access to the PUF with pre- and postprocessing steps [10]. A strong PUF has so many CRPs that an attacker cannot measure all of them during a limited time period. Furthermore, it should be infeasible to build a digital model that would allow an attacker to come up with the right response on a randomly choosen challenge. In authentication applications, a strong PUF has the advantage that the response of the system can be transmitted without any additional security because each CRP is only used once.

A promising candidate for an electrical strong PUF was the class of Arbiter PUFs. Arbiter PUFs generate their responses by exploiting delay information of, e.g., two identical constructed paths, of ICs [8]. Such an Arbiter PUF has a multi-bit input and computes a 1-bit output. The chosen paths are stimulated via multiplexing by the specific challenge and an arbiter compares which of the both competing paths was faster. By concatenating the responses, corresponding to different challenges, a unique key is extracted. Variations of the Arbiter PUF presented in the literature include the XOR Arbiter PUF [8], the Lightweight PUF [40] and the Feed Forward Arbiter PUF [7], which aim for a higher security level than the original Arbiter PUF. However, it has been shown several times that it is possible to model the Arbiter PUFs behavior based on a given set of CRPs by machine learning (ML) techniques, e.g. [41], [42].

Weak PUFs have only a few CRPs, or in some cases, just one. Hence, measures are needed to protect the key against unauthorised access. A popular candidate from this PUF class is the SRAM PUF, introduced by Guajardo et al. [9]. This approach utilizes the power-up behavior of SRAM cells. On power-up the bi-stable memory cells of a SRAM memory tend to either the same bit value with high propability or a random bit. The PUF is formed out of the robust cells, which behave robust on every power-up. The concatenation of the start-up values of all these memory cells is a unique characteristic of each memory array. SRAM-based PUFs can deliver a large number of bits, with the size of an SRAM array as the only limit, and the memory cells do not correlated with each other. Advantageously, SRAM cells are inherent in most semiconductor devices. Hence, it does not require additional devices or modifications in the manufacturing process.

However, it has been already shown that it is possible to read out SRAM PUFs by invasive and semi-invasive attacks [43]. Furthermore, Helfemeier et al. produced a physical clone of a SRAM PUF [44].

Note that weak and strong PUFs aim at different purposes. Strong PUFs could be compared with a physical hash function, whereas weak PUFs are used for safeguard a secret key [45].

Until now, MEMS-based PUFs have received little attention, unlike Arbiter or SRAM PUFs. The first MEMS-based PUF was proposed by Rosenfeld et al. [46]. Their method uses an array of on-chip photodiodes and a translucent coating. The transmittance of the coating is not uniform and causes variations of the measured light level. The key is generated by the variations between the amounts of light sensed by the photodiodes.
Another work focused on MEMS is from Aysu et al. [14]. They used the deviations of an accelerometer's self-test and offset values for a low-cost device authentication. However, they stated that their keys did not achieve the uniqueness as the keys of, e.g., SRAM PUFs.

References

[1] M. Weiser, "The computer for the 21st century-scientific american special issue on communications," *Computers, and Networks (September 1991)*, 1991.

[2] D. Evans, "The internet of things — how the next evolution of the internet is changing everything," *CISCO white paper*, vol. 1, p. 14, 2011.

[3] K. Eldefrawy, G. Tsudik, A. Francillon, and D. Perito, "SMART: secure and minimal architecture for (establishing dynamic) root of trust," in *19th Annual Network and Distributed System Security Symposium, NDSS 2012, San Diego, California, USA, February 5-8, 2012*. The Internet Society, 2012.

[4] F. F. Brasser, B. E. Mahjoub, A. Sadeghi, C. Wachsmann, and P. Koeberl, "Tytan: tiny trust anchor for tiny devices," in *Proceedings of the 52nd Annual Design Automation Conference, San Francisco, CA, USA, June 7-11, 2015*. ACM, 2015, pp. 34:1–34:6.

[5] N. Asokan, F. F. Brasser, A. Ibrahim, A. Sadeghi, M. Schunter, G. Tsudik, and C. Wachsmann, "SEDA: scalable embedded device attestation," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-6, 2015*, 2015, pp. 964–975.

[6] B. Gassend, D. E. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in *Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS 2002, Washington, DC, USA, November 18-22, 2002*, V. Atluri, Ed. ACM, 2002, pp. 148–160.

[7] J. Lee, D. Lim, B. Gassend, G. 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, 2004. Digest of Technical Papers. 2004 Symposium on*, June 2004, pp. 176–179.

[8] G. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Design Automation Conference, 2007. DAC '07. 44th ACM/IEEE*, June 2007, pp. 9–14.

[9] J. Guajardo, S. Kumar, G.-J. Schrijen, and P. Tuyls, “Fpga intrinsic pufs and their use for ip protection,” in *Cryptographic Hardware and Embedded Systems - CHES 2007*, ser. Lecture Notes in Computer Science, P. Paillier and I. Verbauwhede, Eds. Springer Berlin Heidelberg, 2007, vol. 4727, pp. 63–80.

[10] B. Gassend, M. V. Dijk, D. Clarke, E. Torlak, S. Devadas, and P. Tuyls, “Controlled physical random functions and applications,” *ACM Trans. Inf. Syst. Secur.*, vol. 10, no. 4, pp. 3:1–3:22, Jan. 2008.

[11] D. Samyde, S. P. Skorobogatov, R. J. Anderson, and J. Quisquater, “On a new way to read data from memory,” in *Proceedings of the First International IEEE Security in Storage Workshop, SISW 2002, Greenbelt, Maryland, USA, December 11, 2002*. IEEE Computer Society, 2002, pp. 65–69.

[12] S. P. Skorobogatov, “Data remanence in flash memory devices,” in *Cryptographic Hardware and Embedded Systems - CHES 2005, 7th International Workshop, Edinburgh, UK, August 29 - September 1, 2005, Proceedings*, ser. Lecture Notes in Computer Science, J. R. Rao and B. Sunar, Eds., vol. 3659. Springer, 2005, pp. 339–353.

[13] J. A. Halderman, S. D. Schoen, N. Heninger, W. Clarkson, W. Paul, J. A. Calandrino, A. J. Feldman, J. Appelbaum, and E. W. Felten, “Lest we remember: cold-boot attacks on encryption keys,” *Commun. ACM*, vol. 52, no. 5, pp. 91–98, 2009.

[14] A. Aysu, N. F. Ghalaty, Z. Franklin, M. P. Yali, and P. Schaumont, “Digital fingerprints for low-cost platforms using MEMS sensors,” in *Proceedings of the Workshop on Embedded Systems Security*, ser. WESS ’13. New York, NY, USA: ACM, 2013.

[15] Y.-J. Chang, W. Zhang, and T. Chen, “Biometrics-based cryptographic key generation.” in *IEEE International Conference on Multimedia and Expo (ICME)*, vol. 3, 2004.

[16] J. M. G. Linnartz and P. Tuyls, “New shielding functions to enhance privacy and prevent misuse of biometric templates,” in *Audio-and Video-Based Biometrie Person Authentication, 4th International Conference, AVBPA 2003, Proceedings*, ser. LNCS, J. Kittler and M. S. Nixon, Eds., vol. 2688. Springer, June 9-11, 2003, pp. 393–402.

[17] Y. Dodis, L. Reyzin, and A. Smith, “Fuzzy extractors: How to generate strong keys from biometrics and other noisy data,” in *Advances in cryptology-Eurocrypt 2004*. Springer, 2004, pp. 523–540.

[18] C. Acar and A. Shkel, *MEMS vibratory gyroscopes: structural approaches to improve robustness*. Springer Science & Business Media, 2008.

[19] U. Rührmair, S. Devadas, and F. Koushanfar, *Introduction to Hardware Security and Trust*. Springer, 2012, ch. Security Based on Physical Unclonability and Disorder, pp. 65 – 102.

[20] H. Zhun and C. Hongyi, “A truly random number generator based on thermal noise,” in *ASIC, 2001. Proceedings. 4th International Conference on*, 2001, pp. 862–864.

[21] G. Taylor and G. Cox, “Behind intels new random-number generator,” *IEEE Spectrum*, Aug 2011.

[22] M. Wolf and T. Gendrullis, “Design, implementation, and evaluation of a vehicular hardware security module,” in *Information Security and Cryptology - ICISC 2011*, ser. Lecture Notes in Computer Science, H. Kim, Ed. Springer Berlin Heidelberg, 2012, vol. 7259, pp. 302–318.

[23] T. Morris, “Trusted platform module,” in *Encyclopedia of Cryptography and Security*. Springer, 2011, pp. 1332–1335.

[24] F. Schön, “Frequenzgenauigkeit von silizium-basierten mikroelektromechanischen, passiv kompensierten resonatoren für kraftfahrzeuganwendungen,” Ph.D. dissertation, Technische Fakultät der Universität Erlangen-Nürnberg, 2010.

[25] F. M. Willems, Y. M. Shtarkov, and T. J. Tjalkens, “The context-tree weighting method: basic properties,” *Information Theory, IEEE Transactions on*, vol. 41, no. 3, pp. 653–664, 1995.

[26] ——, “Context weighting for general finite-context sources,” *IEEE transactions on information theory*, vol. 42, no. 5, pp. 1514–1520, 1996.

[27] T. Ignatenko, G.-J. Schrijen, B. Skoric, P. Tuyls, and F. Willems, “Estimating the secrecy-rate of physical unclonable functions with the context-tree weighting method,” in *Information Theory, 2006 IEEE International Symposium on*. IEEE, 2006, pp. 499–503.

[28] Y. Gao, I. Kontoyiannis, and E. Bienenstock, “Estimating the entropy of binary time series: Methodology, some theory and a simulation study,” *Entropy*, vol. 10, no. 2, pp. 71–99, 2008.

[29] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, and E. Barker, “A statistical test suite for random and pseudorandom number generators for cryptographic applications,” DTIC Document, Tech. Rep., 2001.

[30] V. Van der Leest, G.-J. Schrijen, H. Handschuh, and P. Tuyls, “Hardware intrinsic security from d flip-flops,” in *Proceedings of the fifth ACM workshop on Scalable trusted computing*. ACM, 2010, pp. 53–62.

[31] E. Barker and J. Kelsey, “Nist draft special publication 800-90b recommendation for the entropy sources used for random bit generation,” 2012.

[32] U. M. Maurer, “A universal statistical test for random bit generators,” *Journal of cryptology*, vol. 5, no. 2, pp. 89–105, 1992.

[33] Y. Dodis, J. Katz, L. Reyzin, and A. Smith, “Robust fuzzy extractors and authenticated key agreement from close secrets,” in *Advances in Cryptology-CRYPTO 2006*. Springer, 2006, pp. 232–250.

[34] A. Van Herrewege, S. Katzenbeisser, R. Maes, R. Peeters, A.-R. Sadeghi, I. Verbauwhede, and C. Wachsmann, “Reverse fuzzy extractors: Enabling lightweight mutual authentication for puf-enabled rfids,” in *Financial Cryptography and Data Security*. Springer, 2012, pp. 374–389.

[35] P. Koeberl, J. Li, A. Rajan, and W. Wu, “Entropy loss in puf-based key generation schemes: The repetition code pitfall,” in *Hardware-Oriented Security and Trust (HOST), 2014 IEEE International Symposium on*. IEEE, 2014, pp. 44–49.

[36] A. Bogdanov, M. Knežević, G. Leander, D. Toz, K. Varıcı, and I. Verbauwhede, “Spongent: A lightweight hash function,” in *Cryptographic Hardware and Embedded Systems–CHES 2011*. Springer, 2011, pp. 312–325.

[37] D. Gugel, “Ordnungsreduktion in der mikrosystemtechnik,” Ph.D. dissertation, TU Chemnitz, 2009.

[38] U. Rührmair, J. Sölter, and F. Sehnke, “On the foundations of physical unclonable functions,” Cryptology ePrint Archive, Report 2009/277, 2009, http://eprint.iacr.org/.

[39] F. Armknecht, R. Maes, A. Sadeghi, O.-X. Standaert, and C. Wachsmann, “A formalization of the security features of physical functions,” in *Security and Privacy (SP), 2011 IEEE Symposium on*, May 2011, pp. 397–412.

[40] M. Majzoobi, F. Koushanfar, and M. Potkonjak, “Lightweight secure pufs,” in *Computer-Aided Design, 2008. ICCAD 2008. IEEE/ACM International Conference on*, Nov 2008, pp. 670–673.

[41] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber, “Modeling attacks on physical unclonable functions,” in *Proceedings of the 17th ACM Conference on Computer and Communications Security*, ser. CCS ’10. New York, NY, USA: ACM, 2010, pp. 237–249.

[42] U. Rührmair, J. Sölter, F. Sehnke, X. Xu, A. Mahmoud, V. Stoyanova, G. Dror, J. Schmidhuber, W. Burleson, and S. Devadas, “Puf modeling attacks on simulated and silicon data,” *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 11, pp. 1876–1891, 2013.

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

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

[45] U. Rührmair and D. Holcomb, “Pufs at a glance.” in *Proceedings - Design, Automation and Test in Europe, DATE*, 2014.

[46] K. Rosenfeld, E. Gavas, and R. Karri, “Sensor physical unclonable functions,” in *Hardware-Oriented Security and Trust (HOST), 2010 IEEE International Symposium on*, June 2010, pp. 112–117.