IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 12, 2022, accepted January 19, 2022, date of publication January 27, 2022, date of current version February 3, 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3146989*

# A New Helper Data Scheme for Soft-Decision Decoding of Binary Physical Unclonable Functions

**ROBERT F. H. FISCHER, (Senior Member, IEEE), AND SVEN MÜELICH**
Institute of Communications Engineering, Ulm University, 89081 Ulm, Germany

Corresponding author: Robert F. H. Fischer (robert.fischer@uni-ulm.de)

This work was supported by the Deutsche Forschungsgemeinschaft (DFG), (German Research Foundation), under Grant FI 982/15-1.

**ABSTRACT** Physical unclonable functions (PUFs) exploit randomness in the hardware for the derivation of cryptographic keys. In the literature, usually the readout is two-level quantized and hard-decision channel decoding is used to stabilize the extracted key. In this paper, we assess soft-decision decoding of binary PUFs. It is well known in the literature on channel coding that soft-decision decoding provides significant gains over hard-decision decoding since reliability information about the symbols is utilized. The PUF readout process is interpreted as digital transmission over a noisy channel, the respective capacity is calculated, and the optimum decoding metric is derived. In addition, we propose an augmented helper data scheme which is suited for soft-decision decoding. This scheme utilizes the fact that operations on the analog readout values are possible, opposed to operations on hard-decided binary symbols in classical PUFs. The security of the new scheme is proven and a possible realization is discussed. The performance is covered by numerical simulations and by applying the scheme to measurement data from FPGA implementations of ring oscillator PUFs.

**INDEX TERMS** Physical unclonable functions, helper data scheme, channel capacity.

## I. INTRODUCTION

Physical unclonable functions (PUFs) are hardware primitives that can be used to securely generate and store cryptographic keys. Randomness that occurs from uncontrollable variations in manufacturing processes of physical objects is exploited to extract a response from the hardware, which classically is a binary sequence that is unique for each PUF. Based on the response, a key can be derived. Since the exploited randomness is static over the object's lifetime, a key can be reproduced at any time. Hence, an implicit key storage is implemented, thereby avoiding additional cost and chip area, and increasing the security compared to a protected non-volatile memory for key storage. Since reproducing the key might be erroneous due to environmental effects like changing temperature or supply voltage, channel coding has to be applied to guarantee stable keys.

Strategies for *channel decoding* differentiate between *hard-decision decoding* and *soft-decision decoding*. In hard-decision decoding, *tentative decisions* are produced by a threshold device (i.e., a quantization operation). Based on these "hard" symbols, the decoder aims to estimate the transmitted codeword. In contrast, in soft-decision decoding reliability information about the symbols is present or can be extracted; these "soft" values are utilized in the decoder. In practical schemes, the reliability information is often expressed as so-called *log-likelihood ratios*.

Traditionally, PUFs employ hard-decision decoding; the PUF readout is quantized (threshold operation) and all further operations are based on these quantized symbols. To enable decoding, during initialization, *helper data* is generated. Via the helper data, the original PUF response is transferred to a codeword of the desired error-correcting code. During reproduction, the readout may differ from the original PUF response. However, applying the helper data, the response is transferred to the form of codeword plus superimposed error word. If the error word has a small enough Hamming weight, decoding will be possible, cf., e.g., [1]–[4]. Note that in the classical setting, responses, helper data, and codewords are all assumed to be binary.

The concept of soft-decision decoding has also been transferred to PUFs. Essentially two methods for gathering soft information from PUF measurements gained interest in the literature. First, reliability information about the individual PUF cells can be obtained by repeatedly extracting the binary PUF response during initialization and evaluating the fraction

The associate editor coordinating the review of this manuscript and approving it for publication was Jin Sha.

of ones for each position. These reliability information can be used to calculate the decoding metric, e.g., [5]–[7], or to improve the channel by only using highly reliable response bits, e.g., [8], [9]. Second, instead of deriving a quantized binary response, depending on the PUF construction, *real numbers* can directly be extracted. For example, real-valued frequency differences of ring oscillators may be exploited, cf. [8].

In the present paper, we follow the second line of work and deal with soft-decision decoding of binary PUFs. To that end, we interpret the PUF readout process as digital transmission over a noisy channel and calculate the respective capacity. In addition, the optimum decoding metric is derived. Based on these information-theoretic considerations, the code rates can be chosen. Moreover, we propose an augmented helper data scheme which is suited for soft-decision decoding. This scheme utilizes the fact that operations on the analog (non-quantized) readout values are possible, opposed to operations on hard-decided binary symbols in classical PUFs. The security of the new scheme is proven and a possible realization is discussed. The performance is covered by numerical simulations and by applying the scheme to measurement data from FPGA implementations of ring oscillator PUFs.

PUFs are usually categorized into *weak* and *strong* PUFs. We primarily address weak PUFs (typically used for key generation, i.e., a unique fingerprint is delivered based on the properties of the hardware), although the discussed concepts can be translated to strong PUFs (for example used for authentication, i.e., the unique response is additionally dependent on a challenge). In addition, we consider the coding/decoding scheme and do not study attacks, as, e.g., done in [10], [11], and do not address countermeasures as, e.g, done in configurable ROPUFs or transformer PUF, cf. [12].

The paper is organized as follows: In Sec. II, PUFs are reviewed and the use of soft information is discussed. The capacity is calculated and the optimum soft-decision decoding metric is derived. A new helper data scheme which exploits the degrees of freedom additionally present when operating on the analog readout is presented in Sec. III. Its security is proven and its free parameters are optimized for achieving best performance. Numerical examples are given. The paper closes in Sec. IV with a brief summary.

## II. PHYSICAL UNCLONABLE FUNCTIONS EMPLOYING SOFT-DECISION DECODING

In this section we review classical PUFs and study those which directly employ an analog quantity which is extracted from the hardware. Based on an information-theoretic analysis—in particular considering the capacity when interpreting the readout process of PUFs as digital transmission over a noisy channel—we compare the potential performance when using hard decisions and soft information, respectively.

### A. RING OSCILLATOR PUFs

PUFs, introduced in [13], exploit intrinsic randomness that occurs due to variations in the manufacturing process of physical items. Since the extracted randomness is usually static over the lifetime of the PUF, keys can be regenerated when required by a cryptosystem, and hence, no non-volatile, protected memory is needed to implement a key storage. Thereby PUFs replace pseudo random number generators and non-volatile memories to provide secure key generation and storage, respectively.

Essentially, the randomness in PUFs is either extracted from delays in electronic components or from the behavior of memory cells. We focus on *ring oscillator PUFs* (ROPUFs), the most prominent member from the group of delay-based PUFs, cf. [14], [15]. A *ring oscillator* (RO) is a loop consisting of an odd number of inverters. If a signal propagates through the RO, it oscillates with a frequency whose actual value depends on the random delays in its inverters and wires. Fig. 1 visualizes the structure of a ROPUF.

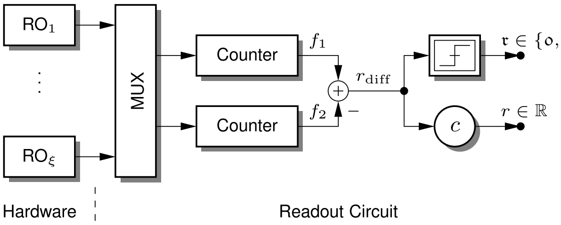

**FIGURE 1. Structure of a ring oscillator PUF with quantized (binary) readout $\mathfrak{r}$ and (suitably scaled) analog readout $r$, respectively.**

In a PUF implementation, pairs of ROs are selected by a multiplexer. The frequencies of the ROs are measured by counters and the frequency difference $r_{\text{diff}}$ is calculated. In the classical description of ROPUFs, depending on the sign of $r_{\text{diff}}$ either the binary symbol[1] $\mathfrak{0}$ or $\mathfrak{1}$ is derived. The symbols of $n$ RO pairs are combined into a vector $\boldsymbol{\mathfrak{r}} = [\mathfrak{r}_1, \ldots, \mathfrak{r}_n]$ which establishes the extracted information.

When re-extracting this information again in a *reproduction phase*, due to variations in the environmental conditions (e.g., temperature or supply voltage), errors might occur, i.e., the word $\boldsymbol{\mathfrak{r}}$ will differ from the word $\boldsymbol{\mathfrak{r}}_{\text{ref}}$ derived during initialization. In the literature, this behavior is traditionally modeled by $\boldsymbol{\mathfrak{r}}_{\text{ref}}$, the nominal/reference readout being transmitted over a *binary symmetric channel* (BSC); its bit error probability is often approximated by $p \approx 0.14$, e.g., [2], [16], [17].

Hence, in order to guarantee a stable result, *channel coding* has to be employed to correct the readout errors. However, the nominal readout $\boldsymbol{\mathfrak{r}}_{\text{ref}}$ is in general not a valid codeword from a given binary channel code. This problem is solved by employing a so-called *helper data scheme* (HDS); the most-often used in practical applications is the *code-offset algorithm* according to [18]–[20].

[1] Notation: We distinguish between quantities from the set of real numbers $\mathbb{R}$ (conventional font) and variables over the binary field $\mathbb{F}_2$ (fraktur font, e.g., $\mathfrak{m}_i = \mathfrak{0}$). In addition, we distinguish between scalars (normal font) and vectors (bold font). Real vectors, e.g., $\boldsymbol{r} = [r_1, \ldots, r_n]^{\mathsf{T}}$, are *column* vectors, however, as usual in channel coding, vectors over the finite field, e.g., $\boldsymbol{\mathfrak{c}} = [\mathfrak{c}_1, \ldots, \mathfrak{c}_n]$, are *row* vectors.

There, in an *initialization phase* which is carried out in a secure environment, a $k$-bit message $\mathfrak{m}$ is randomly, uniformly drawn and encoded (ENC) into a codeword $\mathfrak{c} = \mathsf{ENC}(\mathfrak{m})$ employing a given binary error-correcting code $\mathcal{C}$. The word $\mathfrak{h} = \mathfrak{r}_{\mathrm{ref}} \oplus \mathfrak{c}$ established the helper data ($\oplus$: addition over the binary field $\mathbb{F}_2$); it may be stored publicly. In the *reproduction phase*, the (erroneous) word $\mathfrak{r}$ is extracted from the hardware and $\mathfrak{y} = \mathfrak{r} \oplus \mathfrak{h}$ is calculated. Due to construction, $\mathfrak{y} = \mathfrak{c} \oplus (\mathfrak{r}_{\mathrm{ref}} \oplus \mathfrak{r}) = \mathfrak{c} \oplus \mathfrak{e}$, i.e., the codeword plus an additive error given by the deviation between $\mathfrak{r}_{\mathrm{ref}}$ and $\mathfrak{r}$ is present. If the Hamming weight of $\mathfrak{e}$ is below the error correcting capability, a (hard-decision) channel decoder is able to recover the correct codeword $\mathfrak{c}$ and, thus, the associated message $\mathfrak{m}$.

### B. SOFT-DECISION DECODING

Alternatively, as done in [21], the real-valued frequency differences $r_{\mathrm{diff}}$ can be utilized directly in a *soft-decision decoder*. It is well known in the literature that soft-decision decoding provides significant gains over hard-decision decoding since *reliability information* about the symbols is utilized.

To that end, we suitably normalize the (analog) frequency difference $r_{\mathrm{diff}}$ by multiplying with a scaling factor $c$ (additionally a possible mean is removed). The normalized, real-valued readout symbols are denoted as $r$. As shown in [21] for ROPUFs, the readout vector which combines $n$ PUF cells is very well modeled by

$$\boldsymbol{r} = \boldsymbol{r}_{\mathrm{ref}} + \boldsymbol{e}_{\mathrm{ref}}, \tag{1}$$

where both, the reference/nominal readout $\boldsymbol{r}_{\mathrm{ref}}$ and the error word $\boldsymbol{e}_{\mathrm{ref}}$, are zero-mean Gaussian distributed. W.l.o.g. the normalization factor $c$ can be chosen such that $\boldsymbol{r}_{\mathrm{ref}}$ has unit variance (per element, i.e., $\sigma_x^2 = 1$). The error vector $\boldsymbol{e}_{\mathrm{ref}}$ has variance $\sigma_e^2$ per element. As shown in [21], over a wide range of temperatures the error variance is not larger than $\sigma_e^2 = 0.01$; the corresponding *signal-to-noise ratio* (SNR) is thus at least $10\log_{10}(1/\sigma_e^2) = 20$ dB. We expect the scheme to produce reliable outputs for SNRs larger than this worst case.

The above discussed code-offset algorithm can easily be adapted to the case of soft readout [21]. To that end, we define a *mapping* of the binary (finite-field) symbols "0" and "1" to the real-valued elements of a *binary phase shift-keying* (BPSK) alphabet. The mapping is done according to

$$\mathcal{M}(0) = +1, \quad \mathcal{M}(1) = -1. \tag{2}$$

Since $\mathcal{M}(\mathfrak{c}_1 \oplus \mathfrak{c}_2) = \mathcal{M}(\mathfrak{c}_1) \cdot \mathcal{M}(\mathfrak{c}_2)$, a *homomorphism* between addition over $\mathbb{F}_2$ and multiplication of BPSK symbols exists; the addition of "1" over $\mathbb{F}_2$ is equivalently done by a sign flip (multiplication with $-1$) over the real numbers.

As the sign of the readout represents the extracted information, the sign has to be adjusted such that it matches the sign of the desired codeword $\mathfrak{c}$ which is *mapped* (element-wise) to a BPSK constellation, i.e., $\boldsymbol{a} = \mathcal{M}(\mathfrak{c})$. The sign flip for the entire word can be represented by a *signed identity matrix* $\boldsymbol{S}$ ($\pm 1$ on the main diagonal; zero else). In the initialization phase, this matrix (which is equivalent to $\mathfrak{h}$) is calculated and stored publicly.

In the reproduction phase,

$$\boldsymbol{y} = \boldsymbol{S}\boldsymbol{r} = \boldsymbol{S}\big(\boldsymbol{r}_{\mathrm{ref}} + \boldsymbol{e}_{\mathrm{ref}}\big) = \boldsymbol{x} + \boldsymbol{e}, \tag{3}$$

is calculated. As can be seen, the useful (error-free) signal $\boldsymbol{x} = \boldsymbol{S}\boldsymbol{r}_{\mathrm{ref}}$ is distorted by the error $\boldsymbol{e} = \boldsymbol{S}\boldsymbol{e}_{\mathrm{ref}}$. Still both quantities are zero-mean Gaussian distributed with variances $\sigma_x^2 = 1$ and $\sigma_e^2$, respectively.

### C. SIGNAL/CHANNEL MODEL AND CAPACITY

#### 1) STATISTICS OF THE SIGNALS

We start with the model (3) of the processed PUF readout (after application of the helper data), and aim at deriving the optimum decoding metric and the capacity of the scheme. To that end, in Fig. 2, the probability density function (pdf) of the useful (error-free) PUF readout $x$ is depicted; it is zero-mean, unit variance Gaussian distributed.

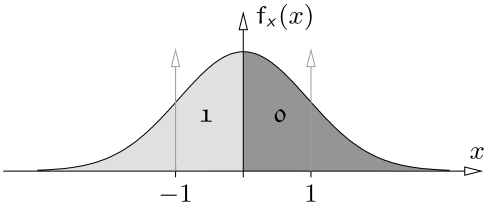

**FIGURE 2.** Regions of the pdf of the PUF readout representing the binary information 0 and 1, respectively. Gray Diracs: Situation in BPSK.

In BPSK, the binary 0 is represented by the real number (signal point) $+1$; the binary 1 by $-1$. Each binary symbol is thus represented by a unique number; the pdf of the BPSK data symbols is *discrete* (gray Diracs in Fig. 2). When considering soft-output PUFs, this unique representation is no longer present. Instead, any number from the region $\mathcal{R}_0 = \{x \mid x \geq 0\}$ represents a binary 0 and any number from the region $\mathcal{R}_1 = \{x \mid x < 0\}$ represents a binary 1, i.e., since both regions are used with probability $1/2$, we have

$$\mathrm{f}_x(x \mid \mathfrak{c} = 0) = \begin{cases} \dfrac{2}{\sqrt{2\pi}}\,\mathrm{e}^{-\frac{x^2}{2}}, & x \in \mathcal{R}_0 = [0, \ -\infty) \\ 0, & \text{else}, \end{cases} \tag{4}$$

$$\mathrm{f}_x(x \mid \mathfrak{c} = 1) = \begin{cases} \dfrac{2}{\sqrt{2\pi}}\,\mathrm{e}^{-\frac{x^2}{2}}, & x \in \mathcal{R}_1 = (-\infty, \ 0) \\ 0, & \text{else}. \end{cases} \tag{5}$$

The binary information is thus represented by any number from a region. The number which is actually present in a PUF cell can be seen as randomly drawn from the regions following the Gaussian distribution within the region.

For performance evaluation, the point of view is thus reversed compared to the operations in a PUF. Instead of stating that the PUF readout is positive or negative and thus gives a 0 or 1, we pretend to communicate a 0 or a 1 and select the actual physical representation randomly from the given region and according to the given statistics. This can be seen as randomness at the transmitter being present, which is an important concept in *physical-layer security* [22].

The selected symbol $x$ is transmitted over an AWGN channel, i.e., zero-mean Gaussian noise with variance $\sigma_e^2$, which is independent of $x$, is superimposed. After some manipulations, the conditional pdfs of the receive signal $y$ given the binary symbol to be communicated can be calculated to be

$$\begin{aligned} \mathsf{f}_y(y \mid \mathfrak{c} = \mathfrak{0}) &= \mathsf{f}_x(y \mid \mathfrak{c} = \mathfrak{0}) * \mathsf{f}_e(y) \\ &= \frac{1}{\sqrt{2\pi\sigma_y^2}}\,\mathrm{e}^{-\frac{y^2}{2\sigma_y^2}}\,\mathrm{Q}(-Fy), \qquad (6) \\ \mathsf{f}_y(y \mid \mathfrak{c} = \mathfrak{1}) &= \mathsf{f}_x(y \mid \mathfrak{c} = \mathfrak{1}) * \mathsf{f}_e(y) \\ &= \frac{1}{\sqrt{2\pi\sigma_y^2}}\,\mathrm{e}^{-\frac{y^2}{2\sigma_y^2}}\,\mathrm{Q}(+Fy), \qquad (7) \end{aligned}$$

where "$*$" denotes convolution and $\mathrm{Q}(x)$ is the *complementary Gaussian integral function*

$$\mathrm{Q}(x) \overset{\text{def}}{=} \int_x^\infty \frac{1}{\sqrt{2\pi}}\,\mathrm{e}^{-\frac{z^2}{2}}\,\mathrm{d}z. \qquad (8)$$

We use the abbreviations

$$\sigma_y^2 \overset{\text{def}}{=} 1 + \sigma_e^2, \qquad F \overset{\text{def}}{=} \frac{1}{\sigma_y\,\sigma_e}. \qquad (9)$$

### 2) CAPACITY

Knowing the pdfs of channel input and output, we are able to calculate the *capacity*—expressed in bit/PUF cell or in short bit/cell—of this channel, i.e., the *mutual information* $\mathrm{I}(\mathfrak{c};\, y)$ between the imagined binary channel input $\mathfrak{c}$ and the real-valued channel output $y$. From basic information theory, e.g., [23], we have (integration is done over the entire real line)

$$\begin{aligned} C_{\mathrm{SD}} &= \mathrm{I}(\mathfrak{c};\, y) \\ &= \int \sum_{\mathfrak{c}\in\{\mathfrak{0},\,\mathfrak{1}\}} \frac{1}{2}\mathsf{f}_y(y \mid \mathfrak{c})\, \log\left(\frac{\mathsf{f}_y(y \mid \mathfrak{c})}{\mathsf{f}_y(y)}\right)\, \mathrm{d}y \\ &= \int \sum_{\mathfrak{c}\in\{\mathfrak{0},\,\mathfrak{1}\}} \frac{1}{2}\mathsf{f}_y(y \mid \mathfrak{c})\, \log\left(\mathsf{f}_y(y \mid \mathfrak{c})\right)\, \mathrm{d}y \\ &\qquad - \int \sum_{\mathfrak{c}\in\{\mathfrak{0},\,\mathfrak{1}\}} \frac{1}{2}\mathsf{f}_y(y \mid \mathfrak{c})\, \log\left(\mathsf{f}_y(y)\right)\, \mathrm{d}y \\ &= \sum_{\mathfrak{c}\in\{\mathfrak{0},\,\mathfrak{1}\}} \frac{1}{2} \int \mathsf{f}_y(y \mid \mathfrak{c})\, \log\left(\mathsf{f}_y(y \mid \mathfrak{c})\right)\, \mathrm{d}y \\ &\qquad - \int \mathsf{f}_y(y)\, \log\left(\mathsf{f}_y(y)\right)\, \mathrm{d}y \\ &= \mathrm{h}(y) - \sum_{\mathfrak{c}\in\{\mathfrak{0},\,\mathfrak{1}\}} \frac{1}{2}\mathrm{h}(y \mid \mathfrak{c}), \qquad (10) \end{aligned}$$

here $\mathrm{h}(\cdot)$ denotes *differential entropy*. Note that, due to symmetry, $\mathsf{f}_y(y \mid \mathfrak{c} = \mathfrak{0}) = \mathsf{f}_y(-y \mid \mathfrak{c} = \mathfrak{1})$ and thus $\mathrm{h}(y \mid \mathfrak{c} = \mathfrak{0}) = \mathrm{h}(y \mid \mathfrak{c} = \mathfrak{1})$. Considering that $e$ is Gaussian distributed and, thus, has differential entropy $\mathrm{h}(e) = \frac{1}{2}\log_2(2\pi \mathrm{e}\sigma_e^2)$, and $y$ is Gaussian distributed with variance $\sigma_y^2 = 1+\sigma_e^2$, we arrive at

$$\begin{aligned} C_{\mathrm{SD}} &= \mathrm{h}(y) - \mathrm{h}(y \mid \mathfrak{c}) \\ &= \mathrm{h}(y) - \mathrm{h}(e) - \mathrm{h}(y \mid \mathfrak{c}) + \mathrm{h}(e) \\ &= \frac{1}{2}\log_2\left(1 + \frac{1}{\sigma_e^2}\right) - \left(\mathrm{h}(y \mid \mathfrak{c}) - \frac{1}{2}\log_2(2\pi \mathrm{e}\sigma_e^2)\right) \\ &= C_{\mathrm{Gauss}} - C_{\text{half Gauss}}, \qquad (11) \end{aligned}$$

where $C_{\mathrm{Gauss}}$ denotes the capacity of the AWGN channel with Gaussian input and $C_{\text{half Gauss}}$ the respective capacity when the input is half-normal distributed. The capacity is thus given as the difference between the capacity when using the entire distribution and that of using only the positive (or negative) half.

### 3) DECODING METRIC

In soft-decision decoding the reliability is often expressed as *log-likelihood ratio* (LLR), which, for equal-probable binary symbols is given by

$$\mathrm{LLR} = \log\left(\frac{\Pr\{\mathfrak{c} = \mathfrak{0} \mid y\}}{\Pr\{\mathfrak{c} = \mathfrak{1} \mid y\}}\right) = \log\left(\frac{\mathsf{f}_y(y \mid \mathfrak{c} = \mathfrak{0})}{\mathsf{f}_y(y \mid \mathfrak{c} = \mathfrak{1})}\right). \qquad (12)$$

Using (6) and (7), we have

$$\mathrm{LLR} = \log\left(\frac{\mathrm{Q}(-F\,y)}{\mathrm{Q}(+F\,y)}\right). \qquad (13)$$

Please note that in case of BPSK over the AWGN channel the LLR would read

$$\mathrm{LLR} = \frac{2}{\sigma_e^2}\,y. \qquad (14)$$

### 4) HARD-DECISION DECODING

For comparison, we also consider hard-decision decoding. Here, the decoder is fed with the hard decisions and operates on the *Hamming metric*.

The hard decisions are characterized by their *bit error ratio* (before decoding). Due to symmetry, the end-to-end model for the channel including the receiver-side quantization is given by *binary symmetric channel* (BSC). Its bit error ratio is given by

$$\mathrm{BER} = \int_0^\infty \mathsf{f}_y(y \mid \mathfrak{c} = \mathfrak{1})\, \mathrm{d}y = \int_{-\infty}^0 \mathsf{f}_y(y \mid \mathfrak{c} = \mathfrak{0})\, \mathrm{d}y. \qquad (15)$$

The capacity of this BSC is then

$$C_{\mathrm{HD}} = 1 - \mathrm{H}_2(\mathrm{BER}), \qquad (16)$$

where the *binary entropy function* is defined as

$$\mathrm{H}_2(x) \overset{\text{def}}{=} -x\log_2(x) - (1-x)\log_2(1-x). \qquad (17)$$

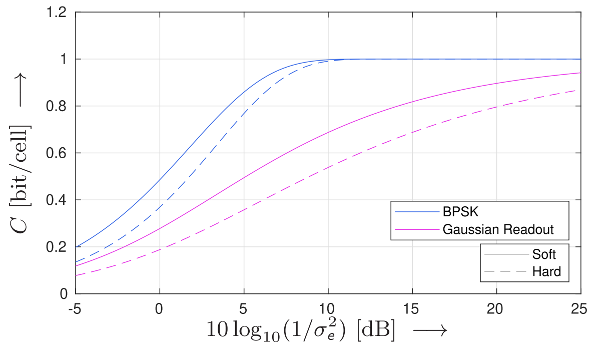

**FIGURE 3.** Capacities over the signal-to-noise ratio (in dB) in case of Gaussian readout and BPSK. Solid: soft decision; Dashed: hard decision.

### D. NUMERICAL EXAMPLE

In Fig. 3, the capacities are plotted over the signal-to-noise ratio. Besides the capacity in case of Gaussian readout, that of BPSK is displayed. The sold lines are valid when utilizing the analog channel output; the dashed lines when only hard decisions are used.

As can be seen, for a fixed SNR, the Gaussian readout provides a much lower capacity than BPSK signaling in conventional digital transmission. Moreover, for a Gaussian readout, hard decision causes a much more significant loss as in case of BPSK. This means, that PUFs utilizing soft-decision decoding enable a large gain over the conventional hard-decision decoding design. For example for a desired capacity (rate of the code) of $C = 0.7$ bit/cell, the curves for Gaussian readout are spaced by approximately 5 dB; soft-decision decoding is possible at 5 dB lower SNR than hard-decision decoding.

## III. NEW HELPER DATA SCHEME

In this section, we present an augmented helper data scheme for soft-decision decoding. The general principle will be enlightened, its security analyzed, and a particular realization is proposed.

### A. BASIC PRINCIPLE

In classical hard-decision binary PUFs the symbols "0" and "1" are flipped in a way such that the noise-free (reference) readout is a randomly chosen valid codeword. Considering the analog readout, as discussed above, a flipping of the sign of the real-valued symbols is the corresponding adequate strategy. However, operating on the analog readout, much more degrees of freedom as for hard-decision binary PUFs are possible.

Consequently, in addition to the sign flipping via the signed identity matrix $\boldsymbol{S}$, we propose a further addition of a suitably chosen real-valued word $\boldsymbol{d} \in \mathbb{R}^n$—subsequently we call this word "dither". In the initialization phase both components are selected and establish the helper data $\mathcal{H} = \{\boldsymbol{S}, \boldsymbol{d}\}$. The situation of the reproduction phase and the interpretation as communication scheme are depicted in Fig. 4.

As in the binary hard-decision case, the component $\boldsymbol{S}$ of the helper data guarantees that $\boldsymbol{r}_\mathrm{c}$ is a valid (mapped) codeword. If the component $\boldsymbol{d}$ would not be present (or $\boldsymbol{d} = \boldsymbol{0}$), the reproduction task is to decode $\boldsymbol{r}_\mathrm{c}$ in additive Gaussian noise $\boldsymbol{e}$. Employing the real-valued word $\boldsymbol{d}$, the useful signal which has to be decoded is $\boldsymbol{x} = \boldsymbol{r}_\mathrm{c} + \boldsymbol{d}$—choosing $\boldsymbol{d}$ suitably, the pdf of $\boldsymbol{x}$ can be shaped and, thus, decoding may be done more reliably.

However, when choosing $\boldsymbol{d}$, two contradicting demands have to be taken into account. On the one hand, the performance of the decoder should be improved. Hence, $\boldsymbol{d}$ has to be dependent on $\boldsymbol{r}_\mathrm{c}$; their sum should be decodable more reliable than only $\boldsymbol{r}_\mathrm{c}$. On the other hand, as the helper data, and thus $\boldsymbol{d}$, is publicly available, it must not reveal any information about $\boldsymbol{r}_\mathrm{c}$.

We first analyze the security of this scheme and then present a possible choice of $\boldsymbol{d}$ which fulfills the contradicting demands.

### B. SECURITY OF THE HELPER DATA SCHEME

As described above, the $k$-bit message $\mathfrak{m}$ is chosen uniformly at random. Encoding and mapping to BPSK symbols is a one-to-one function and gives $\boldsymbol{a} = \mathcal{M}(\mathsf{ENC}(\mathfrak{m}))$. The helper data $\boldsymbol{S}$ is chosen such $\operatorname{sign}(\boldsymbol{S}\boldsymbol{r}_\mathrm{ref}) = \operatorname{sign}(\boldsymbol{r}_\mathrm{c}) = \boldsymbol{a}$ is the valid chosen codeword; the part $\boldsymbol{d}$ of the helper data is generated suitably.

#### a: DECODABILITY

First, it has to be proven that knowing the PUF readout $\boldsymbol{r}_\mathrm{ref}$ (ideal, noise-free case) and the helper data $\mathcal{H} = \{\boldsymbol{S}, \boldsymbol{d}\}$, the message is known. To that end we study the mutual information; the chain rule [23] gives

$$\begin{aligned}\mathrm{I}(\mathfrak{m};\ \boldsymbol{r}_\mathrm{ref}, \{\boldsymbol{S},\boldsymbol{d}\}) &= \mathrm{I}(\mathfrak{m};\ \boldsymbol{r}_\mathrm{ref}, \boldsymbol{S}) + \mathrm{I}(\boldsymbol{m};\ \boldsymbol{d} \mid \boldsymbol{r}_\mathrm{ref}, \boldsymbol{S})\\ &= \mathrm{I}(\mathfrak{m};\ \boldsymbol{r}_\mathrm{c}) + \mathrm{I}(\mathfrak{m};\ \boldsymbol{d} \mid \boldsymbol{r}_\mathrm{c}). \qquad (18)\end{aligned}$$

As the sign of $\boldsymbol{r}_\mathrm{c}$ is a one-to-one deterministic function of $\mathfrak{m}$, we have

$$\mathrm{I}(\mathfrak{m};\ \boldsymbol{r}_\mathrm{c}) = \mathrm{I}(\mathfrak{m};\ \boldsymbol{a}) = k, \qquad (19)$$

$$\begin{aligned}\mathrm{I}(\mathfrak{m};\ \boldsymbol{d} \mid \boldsymbol{r}_\mathrm{c}) &= \mathrm{H}(\mathfrak{m} \mid \boldsymbol{r}_\mathrm{c}) - \mathrm{H}(\mathfrak{m} \mid \boldsymbol{d}, \boldsymbol{r}_\mathrm{c})\\ &= 0 - 0 = 0, \qquad (20)\end{aligned}$$

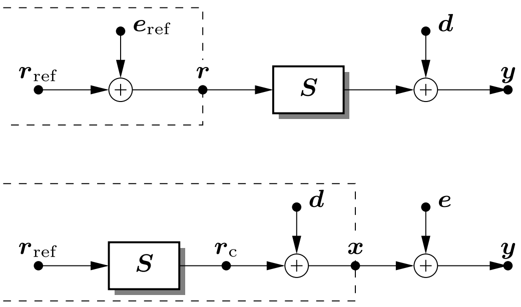

**FIGURE 4.** Block diagram of the processing at the reproduction phase (top) and interpretation as communication scheme (bottom).

thus

$$\mathrm{I}(\mathfrak{m};\ \boldsymbol{r}_{\mathrm{ref}}, \{\boldsymbol{S}, \boldsymbol{d}\}) = k + 0 \ = \ k. \tag{21}$$

### b: NO LEAKAGE WHEN KNOWING THE PUF READOUT

Second, since the message $\mathfrak{m}$ is drawn independently from the PUF readout $\boldsymbol{r}_{\mathrm{ref}}$, by definition, we have

$$\mathrm{I}(\mathfrak{m};\ \boldsymbol{r}_{\mathrm{ref}}) = 0. \tag{22}$$

This means that no information about $\mathfrak{m}$ can be extracted when only $\boldsymbol{r}_{\mathrm{ref}}$ is known.

### c: NO LEAKAGE WHEN KNOWING THE HELPER DATA

Finally, we consider the case when only the (public) helper data $\mathcal{H} = \{\boldsymbol{S}, \boldsymbol{d}\}$ is known. Since $\mathfrak{m}$ determines $\boldsymbol{a} = \mathrm{sign}(\boldsymbol{r}_{\mathrm{c}})$, we have

$$\begin{aligned} \mathrm{I}(\mathfrak{m};\ \{\boldsymbol{S}, \boldsymbol{d}\}) &= \mathrm{I}(\mathfrak{m};\ \boldsymbol{S}) + \mathrm{I}(\mathfrak{m};\ \boldsymbol{d} \mid \boldsymbol{S}), \\ &= \mathrm{I}(\boldsymbol{a};\ \boldsymbol{S}) + \mathrm{I}(\boldsymbol{a};\ \boldsymbol{d} \mid \boldsymbol{S}) \\ &= \mathrm{I}(\boldsymbol{a};\ \boldsymbol{S}) + \mathrm{I}(\boldsymbol{a};\ \boldsymbol{d}). \end{aligned} \tag{23}$$

The last equation is valid since $\boldsymbol{S}$ is irrelevant when knowing $\boldsymbol{a}$. Hence, we have the intuitive result that a possible leakage is the sum of the leakage via $\boldsymbol{S}$ and that via $\boldsymbol{d}$. We abbreviate the latter by $\mathrm{I}(\boldsymbol{a};\ \boldsymbol{d}) = n\,L_{\mathrm{d}}$ and calculate $L_{\mathrm{d}}$ when discussing a specific way to choose $\boldsymbol{d}$.

The former mutual information can be written as

$$\mathrm{I}(\boldsymbol{a};\ \boldsymbol{S}) = \mathrm{H}(\boldsymbol{S}) - \mathrm{H}(\boldsymbol{S} \mid \boldsymbol{a}). \tag{24}$$

The entropy of a signed identity matrix is $\mathrm{H}(\boldsymbol{S}) = \log_2(2^n) = n$. Knowing $\boldsymbol{a}$ (but not $\boldsymbol{r}_{\mathrm{ref}}$) does not decrease the uncertainty about $\boldsymbol{S}$. Formally, $\mathrm{H}(\boldsymbol{S} \ \mid \ \boldsymbol{a}) \ = \ n$. Hence, combining the results yields

$$\mathrm{I}(\boldsymbol{a};\ \boldsymbol{S}) = 0, \tag{25}$$

and finally

$$\mathrm{I}(\mathfrak{m};\ \{\boldsymbol{S}, \boldsymbol{d}\}) = n\,L_{\mathrm{d}}. \tag{26}$$

When the leakage $L_{\mathrm{d}}$ due to the word $\boldsymbol{d}$ can be made zero, the entire leakage of the scheme will be zero.

## C. SCHEME WITH DITHER

We now present a first approach for choosing the word $\boldsymbol{d}$. To that end, two extreme cases can be observed.

First, if $\boldsymbol{d} = \boldsymbol{0}$, regardless of the other quantities, no modification of $\boldsymbol{r}_{\mathrm{c}}$ is done to obtain $\boldsymbol{x}$. Thus, $\boldsymbol{x}$ is Gaussian distributed and the initial situation is present. Clearly, no leakage is caused and $L_{\mathrm{d}} = 0$.

Second, if $\boldsymbol{d} = \mathrm{sign}(\boldsymbol{r}_{\mathrm{c}}) - \boldsymbol{r}_{\mathrm{c}}$ is chosen, we have $\boldsymbol{x} = \boldsymbol{r}_{\mathrm{c}} + \boldsymbol{d} = \mathrm{sign}(\boldsymbol{r}_{\mathrm{c}}) \in \{\pm 1\}^n$; the situation of a BPSK transmission would be present. However, here a significant leakage is caused. In general, error-free decoding of the message $\mathfrak{m}$ purely based on $\boldsymbol{d}$ would be possible.

Consequently, a suited strategy has to be in between these extreme cases— $\boldsymbol{x}$ should be moved away from a Gaussian distribution to support decoding, but at the same time no (or only little) leakage should be caused. In other words, $\boldsymbol{d}$ should be selected based on $\boldsymbol{r}_{\mathrm{ref}}$, $\boldsymbol{S}$, and $\boldsymbol{a}$, such that the performance of the reconstruction is increased but the leakage is as small as possible. A possible procedure is to maximize the *capacity* of the AWGN channel with input $\boldsymbol{x}$ minus the *leakage* caused by the knowledge of $\boldsymbol{d}$.

### 1) DITHER AND PROBABILITY DENSITY FUNCTIONS

Our proposal is as follows: As already done in [21], the sign of the readout $\boldsymbol{r}_{\mathrm{ref}}$ is flipped, i.e., $\boldsymbol{r}_{\mathrm{c}} = \boldsymbol{S}\,\boldsymbol{r}_{\mathrm{ref}}$ is generated ($\boldsymbol{S}$ is selected suitably), such that $\mathrm{sign}(\boldsymbol{r}_{\mathrm{c}})$ is a valid (mapped) codeword.

The new part $\boldsymbol{d}$ is generated as follows; all subsequent calculations are done individually per elements of the word $\boldsymbol{r}_{\mathrm{c}}$. Let $u$ be a random variable, independent of all other variables, and uniformly distributed over the interval $[0, \ \ \mu]$, where $\mu$ is a free parameter to be optimized. Further, let $\nu$ be a given threshold which has to be optimized, too. Then, the considered element $d$ of the vector $\boldsymbol{d}$ is calculated as follows

$$d = \begin{cases} u, & r_{\mathrm{c}} < -\nu \text{ and } 0 \le r_{\mathrm{c}} < \nu \\ -u, & -\nu \le r_{\mathrm{c}} < 0 \text{ and } \nu \le r_{\mathrm{c}}, \end{cases}$$

which means that

$$x = r_{\mathrm{c}} + d \ = \begin{cases} r_{\mathrm{c}} + u, & r_{\mathrm{c}} < -\nu \text{ and } 0 \le r_{\mathrm{c}} < \nu \\ r_{\mathrm{c}} - u, & -\nu \le r_{\mathrm{c}} < 0 \text{ and } \nu \le r_{\mathrm{c}}. \end{cases}$$

We call $\boldsymbol{d}$ "dither" as the word $\boldsymbol{r}_{\mathrm{c}}$ is dithered; the elements of $\boldsymbol{x}$ jitter around the original values of $\boldsymbol{r}_{\mathrm{c}}$. The use of a dither is a well-known concept in digital transmission and channel decoding, e.g., [24], [25].

The support of the joint pdf of $r_{\mathrm{c}}$ and $d$ is sketched in Fig. 5. This joint pdf has a Gaussian marginal pdf over $r_{\mathrm{c}}$ and a uniform marginal pdf (interval $[-\mu, \ \ \mu]$) over $d$.

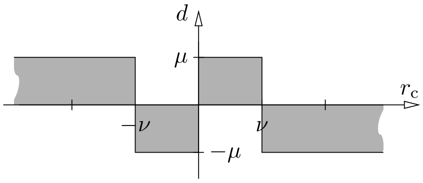

**FIGURE 5.** Support of the joint pdf of $r_{\mathrm{c}}$ and $d$. Shown for $\mu < \nu$.

Please note that

$$\begin{aligned} \Pr\{r_{\mathrm{c}} \ge 0 \mid d < 0\} &= \Pr\{r_{\mathrm{c}} < 0 \mid d \ge 0\} \\ &= 2\,\mathrm{Q}(\nu), \end{aligned} \tag{27}$$

$$\begin{aligned} \Pr\{r_{\mathrm{c}} < 0 \mid d < 0\} &= \Pr\{r_{\mathrm{c}} \ge 0 \mid d \ge 0\} \\ &= 1 - 2\,\mathrm{Q}(\nu). \end{aligned} \tag{28}$$

When knowing the sign of $d$, the entropy of the sign of $r_{\mathrm{c}}$ is thus only $\mathrm{H}_2(2\,\mathrm{Q}(\nu))$. Hence, the leakage (per symbol) caused by knowing $d$ is

$$L_{\mathrm{d}} = 1 - \mathrm{H}_2(2\,\mathrm{Q}(\nu)). \tag{29}$$

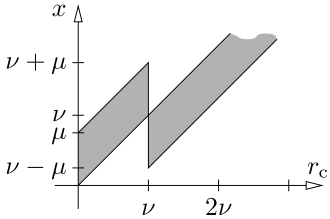

**FIGURE 6.** Support of the joint pdf of $r_c$ and $x$. Shown for $\mu < \nu$. (Only the positive part is shown. The pdf is point-symmetric about the origin.) .

If $\nu = 0.674$, we have $2\,\mathrm{Q}(0.674) = 1/2$; here, knowing $d$ does not give any information about the sign of $r_c$ which carries the message; no leakage is caused.

The support of the joint pdf of $r_c$ and $x = r_c + d$ is sketched in Fig. 6. As above, this joint pdf has a Gaussian marginal pdf over $r_c$ and (for fixed $r_c$) is uniform in $x$ direction.

Integrating this joint pdf over $r_c$ gives the pdf of $x$ (marginal pdf). In anticipating the subsequent results, we may restrict ourselves to the range $\nu/2 \le \mu \le \nu$. For this case, the marginal pdf is given by (for $x \ge 0$ and $\mathrm{f}_x(-x) = \mathrm{f}_x(x)$)

$$\mathrm{f}_x(x) = \begin{cases} \frac{1}{\mu}\big(\mathrm{Q}(0) - \mathrm{Q}(x)\big), & 0 \le x < \nu-\mu \\ \frac{1}{\mu}\big(\mathrm{Q}(0) - \mathrm{Q}(x) + \mathrm{Q}(\nu) - \mathrm{Q}(x+\mu)\big), & \\ & \nu-\mu \le x < \mu \\ \frac{1}{\mu}\big(\mathrm{Q}(x-\mu) - \mathrm{Q}(x) + \mathrm{Q}(\nu) - \mathrm{Q}(x+\mu)\big), & \\ & \mu \le x < \nu \\ \frac{1}{\mu}\big(\mathrm{Q}(x-\mu) - \mathrm{Q}(\nu) + \mathrm{Q}(x) - \mathrm{Q}(x+\mu)\big), & \\ & \nu \le x < \nu+\mu \\ \frac{1}{\mu}\big(\mathrm{Q}(x) - \mathrm{Q}(x+\mu)\big), & x \ge \nu+\mu \end{cases} \tag{30}$$

The influence of the dither on the pdf of $x$ is visualized in Fig. 7. Via the dither, the Gaussian pdf of the readout (magenta) is driven towards the discrete pdf of a BPSK transmit signal (blue). Since the pdf has much less contributions around the threshold $x = 0$, better performance can be expected.

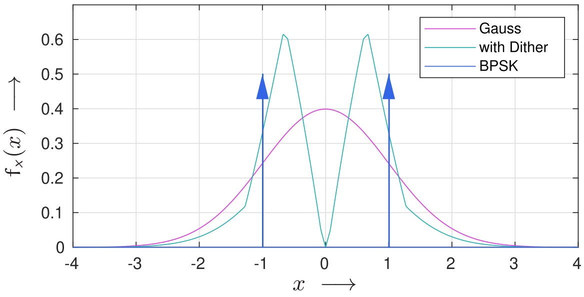

**FIGURE 7.** Pdf of the transmit signal: Gaussian readout, BPSK, and scheme with dither ($\nu = 0.674$, $\mu = 0.6$).

### 2) OPTIMIZATION OF THE PARAMETERS

For best performance, the free parameters $\mu$ (amplitude) and $\nu$ (threshold) have to be optimized. If zero leakage is requested, $\nu = 0.674$ has to be selected but still $\mu$ has to be adjusted.

The optimization can be done as follows: Given the pair $\mu$ and $\nu$, the pdf of the transmit signal $x$ is calculated via (30). Having $\mathrm{f}_x(x)$, the capacity $C_{\mathrm{SD}}$ can be calculated numerically via the procedure explained in Sec. II-C by replacing the Gaussian pdf by the given one. Finally, considering the leakage for the specific choice of $\nu$, the useful capacity $C = C_{\mathrm{SD}} - n\,L_{\mathrm{d}}$ is obtained. For each desired useful capacity (equal to the rate of the code), the optimum pair $\mu$ and $\nu$ (for which the required SNR is minimum) can be determined.

The results are depicted in Fig. 8. There, the capacity in case of Gaussian readout is compared with the case of employing a dither. It is visible, that via a dither, significant gains can be achieved. For example, a desired capacity is $C = 0.7$ bit/cell is already guaranteed at an approximately 4 dB lower SNR. If $\nu = 0.674$ is fixed such that no leakage is present, some (small) loss compared to the case when a leakage is allowed (but the useful capacity is maximized) has to be accepted. However, even the case of no leakage shows very good performance. For reference, the capacity curve of BPSK over the AWGN channel is shown as well. Using a dither, approximately half the distance between Gaussian readout and BPSK can be bridged.

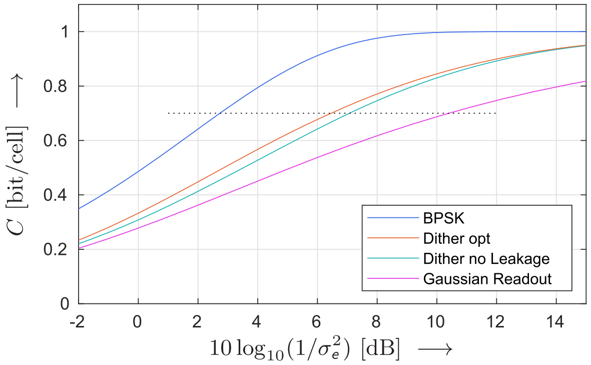

**FIGURE 8.** Comparison of the capacities over the signal-to-noise ratio (in dB) for Gaussian readout, BPSK, and scheme with dither ("opt": $\mu$ and $\nu$ optimized for maximum useful capacity; "no leakage": $\nu = 0.674$ and $\mu$ optimized).

### 3) OPTIMIZATION OF THE PDF

Up to now, the elements of the dither $\boldsymbol{d}$ are uniformly distributed. Obviously, the pdf of the initial random variable $u$ can be optimized. Please note that the above statements on the leakage are valid regardless of the pdf of $u$. Hence, as long as $\nu = 0.674$, no leakage is caused. For optimizing the pdf, we restrict ourselves to this case.

We now allow the pdf of $u$ to follow any function over the interval $[0,\ \mu]$ given by a polynomial of degree $p_{\mathrm{u}}$, i.e., $\mathrm{f}_u(u) = \sum_{l=0}^{p_{\mathrm{u}}} \zeta_l\, u^l$. Thereby, the coefficients $\zeta_l$ have to be normalized such that $\int_0^\mu \mathrm{f}_u(u)\,\mathrm{d}u = 1$. Unfortunately, in this case no closed-form expression for $\mathrm{f}_x(x)$ can be given. However, via numerical optimization the coefficients $\zeta_l$ have been optimized for polynomials up to degree $p_\mathrm{u} = 4$ to maximize capacity. Over a wide range of SNRs, the optimum is very close to a *triangular distribution* $\mathrm{f}_u(u) = \frac{2}{\mu}u$, for $u \in [0,\ \mu]$, and zero else.

#### 4) NUMERICAL SIMULATIONS

Finally, we present results from numerical simulations. *Polar codes* [26] are used as channel coding schemes as a low-complexity soft-input decoding algorithm is available, which can be efficiently implemented in hardware [27]. A code with rate $R = 0.7$ and codelength $n = 1024$ is presumed; the code is designed based on the Bhattacharyya parameter [26], [28]; for the rate-0.7 code we use the design SNR 5.74 dB (3 dB above the capacity limit of BPSK over the AWGN channel; blue curve in Fig. 8). *Successive cancellation decoding* is employed. LLRs clipping to a maximum magnitude of 100 is active.

We plot the *word error ratio* (WER), i.e., the probability that the reproduced message $\hat{\mathfrak{m}}$ differs from the actual message $\mathfrak{m}$. As common for FPGA PUFs, a WER below $10^{-6}$ is desired.

Fig. 9 compares the different situations. First, Gaussian readout without additional dither is considered (magenta). The solid lines are valid for the (correct) LLR calculation according to (13) and the dashed lines are valid for the LLR calculation (14) which is optimum for BPSK. Clearly, the LLR calculation matched to the specific situation gives better results than that for BPSK. However, the loss due to the much simpler calculation is not too large. For comparison (black), the WER when using hard decisions is given. The significant gain due to soft decisions is clearly visible.

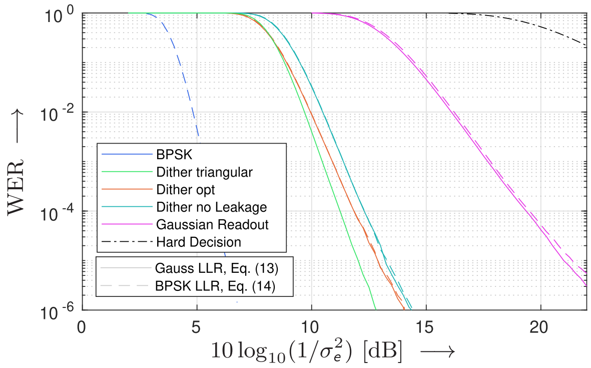

**FIGURE 9. Word error ratio (WER) over the signal-to-noise ratio (in dB). Polar code with rate $R = 0.7$. Uniformly distributed dither (no leakage case: $\nu = 0.674$ and $\mu = 0.6$; leakage case (opt): $\nu = 0.849$ and $\mu = 0.807$; coderate increased to $R = 0.732$) and triangularly distributed dither (no leakage case: $\nu = 0.674$ and $\mu = 0.53$).**

Employing the (uniformly distributed) dither, much better performance can be achieved. The cyan lines are valid for the parameters $\nu = 0.674$ and $\mu = 0.6$; here no leakage is present. The red lines hold for $\nu = 0.849$ and $\mu = 0.807$ (opt); as here a leakage of $1 - \mathrm{H}_2(2\mathrm{Q}(0.849)) = 0.032$ bit/symbol is present, the coderate is increased to $R = 0.732$. Even with this larger coderate, a better performance is achieved. Both LLR calculations (which here are both approximations) give almost the same performance.

The green curve is valid for the case of a triangularly distributed dither (here only the LLR calculation according to (13) has been used). Here, $\nu = 0.674$ (no leakage) is chosen and $\mu = 0.53$ results from the optimization. Employing this optimized pdf of the dither, a gain of approximately 1 dB can be achieved over the situation when the dither is uniformly distributed.

Finally, the potential performance of BPSK over an AWGN channel is shown (blue). Please notice that the relations predicted by the capacity arguments (cf. Fig. 8) are reflected in the word error rate curves. Due to the finite (relatively short) codelength, the absolute gaps are larger than the differences in capacity.

### D. EVALUATION WITH ROPUF MEASUREMENT DATA

At the Institute of Microelectronics at Ulm University, 22 instances of FPGA ROPUFs have been implemented, cf. [15]. Out of the available ROs, $n = 1024$ disjoint pairs have been selected randomly. Each pair has been measured at various temperatures; we use the measurements from $-10$ °C to 50 °C (in steps of 10 °C). Temperature variations, voltage variations, and aging are the most relevant items for readout deviations/errors. However, in contrast to the environmental temperature, the supply voltage can (and will) be stabilized by voltage regulators.

The reference readout $\boldsymbol{r}_\mathrm{ref}$ of each PUF instance is obtained by averaging 10 readouts at a temperatures of 20 °C. For each instance, the message $\mathfrak{m}$ is randomly selected and the helper data (sign-flipping matrix $\boldsymbol{S}$ and uniform dither $\boldsymbol{d}$) is generated as detailed in the present paper. We restrict ourselves to the no-leakage case $\nu = 0.674$). Polar codes with codelength $n = 1024$ are employed.

For verification, 10,000 readouts per PUF instance and per temperature are used. The helper data is applied to the verification readout and decoding is performed.

In Tab. 1, the number of erroneous PUF instance among the 22 instances and the number of word errors per erroneous instance are tabulated. The rate is chosen as $R = 0.7$ (with the optimum choice $\mu = 0.60$), $R = 0.8$ (with $\mu = 0.57$), and $R = 0.9$ bit/cell (with $\mu = 0.55$). Results for the scheme with and without dither are given. For example, for $R = 0.8$ and the scheme with dither, error occurred in 3 out of the 22 instances; 19 instances were free of errors over the entire temperature range. The 3 instances with errors showed a singe or two errors over the entire temperature range (7 temperatures) and all 10,000 readouts per temperature.

The improvement by using the proposed dither is clearly visible. Without dither, in all cases errors occur. For $R = 0.7$, the scheme with dither is able to deliver all messages free of errors. For higher rates, the scheme without dither completely fails; in the scheme with dither only rare errors

**TABLE 1.** Number of erroneous PUF instance (out of the 22 instances and over the entire temperature range) and word errors per erroneous instance for the binary scheme with and without dither.

| | $R = 0.7$ | $R = 0.8$ | $R = 0.9$ |
|---|---|---|---|
| without dither | 7 instances<br>1 to 11 errors | 15 instances<br>1 to 971 errors | 22 instances<br>8272 to 53541 errors |
| with dither | 0 instances<br>— | 5 instances<br>1 to 9 errors | 7 instances<br>1 to 15 errors |

occur. In summary, the proposed scheme is able to operate reliably over a wide range of temperatures and with rates up to $R = 0.7$ bit/cell.

## IV. CONCLUSION

In this paper, soft-decision decoding in binary PUFs has been addressed. By interpreting the PUF readout process as digital transmission over a noisy channel with a specific (uncommon) pdf of the useful signal, the respective capacity has been derived. Moreover, the optimum decoding metric (in form of LLRs) has been given. In addition, a helper data scheme suited for soft-decision decoding has been studied. In particular, an augmentation by an additive dither word has been proposed. The security of this new approach has been proven. The performance is covered by numerical simulations and by evaluating measurement data from FPGA implementations of ring oscillator PUFs. Employing the scheme with dither, rates up to 0.7 bit/cell can be extracted reliably, which is a tremendous gain over state-of-the-art binary PUFs utilizing hard-decision decoding.

Even though we started from a ROPUF, the discussed principles can be applied to any PUF architecture where the analog source of randomness is accessible. In many situations, Gaussian signal and error models are reasonable assumptions in view of the law of large number.

## ACKNOWLEDGMENT

The authors would like to thank Holger Mandry and Maurits Ortmanns, Institute of Microelectronics, Ulm University, for valuable discussions and providing the measurement data of their implementation of ROPUFs.

## REFERENCES

[1] C. Bösch, J. Guajardo, A.-R. Sadeghi, J. Shokrollahi, and P. Tuyls, "Efficient helper data key extractor on FPGAs," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.*, Washington, DC, USA 2008, pp. 181–197.

[2] R. Maes, A. Van Herrewege, and I. Verbauwhede, "PUFKY: A fully functional PUF-based cryptographic key generator," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.*, Leuven, Belgium, 2012, pp. 302–319.

[3] S. Müelich, S. Puchinger, M. Bossert, M. Hiller, and G. Sigl, "Error correction for physical unclonable functions using generalized concatenated codes," 2014, *arXiv:1407.8034*.

[4] S. Puchinger, S. Müelich, M. Bossert, M. Hiller, and G. Sigl, "On error correction for physical unclonable functions," in *Proc. 10th Int. ITG Conf. Syst., Commun. Coding*, Hamburg, Germany, 2015, pp. 1–6.

[5] R. Maes, P. Tuyls, and I. Verbauwhede, "A soft decision helper data algorithm for SRAM PUFs," in *Proc. IEEE Int. Symp. Inf. Theory*, Seoul, South Korea, Jun. 2009, pp. 2101–2105.

[6] R. Maes, P. Tuyls, and I. Verbauwhede, "Low-overhead implementation of a soft decision helper data algorithm for SRAM PUFs," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.*, Lausanne, Switzerland, 2009, pp. 332–347.

[7] M. Taniguchi, M. Shiozaki, H. Kubo, and T. Fujino, "A stable key generation from PUF responses with a fuzzy extractor for cryptographic authentications," in *Proc. IEEE 2nd Global Conf. Consum. Electron. (GCCE)*, Tokyo, Japan, Oct. 2013, pp. 525–527.

[8] M.-D. Yu and S. Devadas, "Secure and robust error correction for physical unclonable functions," *IEEE Des. Test Comput.*, vol. 27, no. 1, pp. 48–65, Jan./Feb. 2010.

[9] M. Hiller, M. Weiner, L. R. Lima, M. Birkner, and G. Sigl, "Breaking through fixed PUF block limitations with differential sequence coding and convolutional codes," in *Proc. 3rd Int. Workshop Trustworthy Embedded Devices*, Berlin, Germany, 2013, pp. 43–54.

[10] J. Shi, Y. Lu, and J. Zhang, "Approximation attacks on strong PUFs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 10, pp. 2138–2151, Oct. 2019.

[11] J. Zhang, C. Shen, H. Su, M. T. Arafin, and G. Qu, "Voltage over-scaling-based lightweight authentication for IoT security," *IEEE Trans. Comput.*, vol. 71, no. 2, pp. 323–336, Feb. 2022.

[12] Z. Wei, Y. Cui, Y. Chen, C. Wang, C. Gu, and W. Liu, "Transformer PUF: A highly flexible configurable RO PUF based on FPGA," in *Proc. IEEE Workshop Signal Process. Syst. (SiPS)*, Oct. 2020, pp. 1–6.

[13] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions," *Science*, vol. 297, no. 5589, pp. 2026–2030, Sep. 2002.

[14] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in *Proc. 9th ACM Conf. Comput. Commun. Secur.*, Washington, DC USA, 2002, pp. 148–160.

[15] A. Herkle, H. Mandry, J. Becker, and M. Ortmanns, "In-depth analysis and enhancements of RO-PUFs with a partial reconfiguration framework on Xilinx Zynq-7000 SoC FPGAs," in *Proc. Int. Symp. Hardw. Oriented Secur. Trust*, McLean, VA, USA, 2019, pp. 238–247.

[16] S. Müelich, S. Puchinger, M. Bossert, M. Hiller, and G. Sigl, "Error correction for physical unclonable functions using generalized concatenated codes," in *Proc. Int. Workshop Algebr. Combinatorical Coding Theory*, Svetlogorsk, Russia, 2014, pp. 1–6.

[17] S. Puchinger, S. Müelich, M. Bossert, M. Hiller, and G. Sigl, "On error correction for physical unclonable functions," in *Proc. 10th Int. ITG Conf. Syst., Commun. Coding*, Hamburg, Germany, 2015, pp. 1–6.

[18] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in *Proc. 6th ACM Conf. Comput. Commun. Secur.*, Singapore, 1999, pp. 28–36.

[19] J.-P. Linnartz and P. Tuyls, "New shielding functions to enhance privacy and prevent misuse of biometric templates," in *Proc. Int. Conf. Audio-Video-Based Biometric Person Authentication*, Guildford, U.K., 2003, pp. 393–402.

[20] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, Interlaken, Switzerland, 2004, pp. 523–540.

[21] S. Muelich, H. Mandry, M. Ortmanns, and R. F. H. Fischer, "A multilevel coding scheme for multi-valued physical unclonable functions," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 3814–3827, 2021.

[22] M. Bloch and J. Barros, *Physical-Layer Security: From Information Theory to Security Engineering*. Cambridge, U.K.: Cambridge Univ. Press, 2011.

[23] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, NY, USA: Wiley, 1991.

[24] G. D. Forney, "On the role of MMSE estimation in approaching the information-theoretic limits of linear Gaussian channels: Shannon meets Wiener," in *Proc. 41st Annu. Allerton Conf. Commun., Control Comput.*, Monticello, IL, USA, Oct. 2003, pp. 430–439.

[25] G. D. Forney, Jr. and G. Ungerboeck, "Modulation and coding for linear Gaussian channels," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2384–2415, Oct. 1998.

[26] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.

[27] A. A. Andi and O. Gazi, "Fast decoding of polar codes using tree structure," *IET Commun.*, vol. 13, no. 14, pp. 2063–2068, Aug. 2019.

[28] H. Vangala, E. Viterbo, and Y. Hong, "A comparative study of polar code constructions for the AWGN channel," 2015, *arXiv:1501.02473*.

**ROBERT F. H. FISCHER** (Senior Member, IEEE) received the Dr.-Ing. and Habilitation degrees from the University of Erlangen–Nürnberg, Erlangen, Germany, in 1996 and 2001, respectively.

From 1992 to 1996, he was a Research Assistant with the Telecommunications Institute, University of Erlangen–Nürnberg. In 1997, he was with the IBM Research Laboratory, Zürich, Switzerland. In 1998, he returned to the University of Erlangen–Nürnberg. In 2005, he spent a sabbatical with ETH Zürich, Zürich. Since 2011, he has been a Full Professor with Ulm University, Ulm, Germany. He is currently teaching the undergraduate and graduate courses on signals and systems and digital communications. He has authored the textbook *Precoding and Signal Shaping for Digital Transmission* (Wiley, 2002). His current research interests include fast, reliable, and secure digital transmission, including single-carrier and multicarrier modulation techniques, information theory, coded modulation, digital communications, signal processing, and especially precoding and shaping techniques.

Dr. Fischer was a recipient of the Dissertation Award from the Technische Fakultät, University of Erlangen–Nürnberg, in 1997, the Publication Award of the German Society of Information Techniques, in 2000, the Wolfgang Finkelnburg Habilitation Award, in 2002, and the Philipp–Reis–Preis Award, in 2005.

**SVEN MÜELICH** received the B.Sc. degree in computer science from Aalen University, Germany, in 2011, and the M.Sc. degree in computer science from Ulm University, in 2013. In 2013, he joined the Institute of Communications Engineering, Ulm University, as a Research Assistant, aiming to gain the Ph.D. degree. His research focus on physical unclonable functions, while he also worked on rank metric codes and code-based cryptography. He defended his Ph.D. thesis with the title "Channel Coding for Hardware-Intrinsic Security," in 2019, and subsequent to his Ph.D. graduation. He has been engaged as a Postdoctoral Researcher at the Institute of Communications Engineering, Ulm University. His main research interest includes coding for physical unclonable functions.