On the Key Generation Rate of Physically Unclonable Functions

Yitao Chen, Muryong Kim, and Sriram Vishwanath
University of Texas at Austin
Austin, TX 78701 USA
Email: [yitaochen, muryong]@utexas.edu, sriram@ece.utexas.edu

Abstract—In this paper, an algebraic binning based coding scheme and its associated achievable rate for key generation using physically unclonable functions (PUFs) is determined. This achievable rate is shown to be optimal under the generated-secret (GS) model for PUFs. Furthermore, a polar code based polynomial-time encoding and decoding scheme that achieves this rate is also presented.

I. INTRODUCTION

Physically unclonable functions (PUFs) form a promising innovative primitive that are increasingly gaining traction in the domains of authentication and secret key storage [1]–[3]. Instead of storing secrets in digital memory, PUFs derive a secret from the physical characteristics of an integrated circuit (IC) that form an inherent part of the device. Such a PUF can be obtained, as even though the mask and manufacturing process is relatively similar among ICs built for a particular purpose, each IC is actually unique due to normal manufacturing variability.

This unique behavior after manufacturing stems from a static randomness due to technological dispersion. This static randomness was characterized by Pelgrom [4], and is known to follow a normal distribution. Unfortunately, PUF outputs are also subject to dynamic randomness due to measurement noise, which is detrimental to the reliability of a PUF as a source for cryptographic elements.

In this paper, we understand the information theoretic limits of key generation using PUFs, given this static and dynamic randomness in the system. As discussed in [1], [2], one of the central use-cases for PUFs is secret key generation, where this key is subsequently utilized in a variety of cryptographic algorithms. A higher key generation rate implies greater security guarantees for the overall system, and therefore, our focus is to understand its limits, and to characterize coding schemes that approach these limits.

A. Related Work and Our Contributions

There is already a considerable body of work on combining PUFs with error correction coding schemes to obtain reliable keys or secrets [3]. Conventionally, these have combined BCH/RS codes with PUFs. More recently, [8] presents simulation results on the combination of a polar code with a PUF, setting the stage for such a combination to be understood analytically. In parallel work to this paper, [9] uncovers the connection between PUF key generation problem and Wyner-Ziv problem [15], and studies a nested polar codes construction scheme based on [10].

In this work, we present a PUF key generation scheme based on a previously well studied model called generated-secret (GS) model. In [6], the authors present the region of achievable secret-key vs. privacy-leakage (key vs. leakage) rates for the GS model. In this paper, we show that the optimal key generation rate is achievable using algebraic binning with linear codes, and uncover the relation between PUF key generation problem and Slepian-Wolf problem. Further, we present encoding and decoding algorithms using polar codes that achieve this rate. Finally, we present simulation results to showcase the performance of our scheme.

Compared to existing literature, we find that our scheme results in a relatively straightforward interpretation of the PUF key generation problem, and results in a key generation rate that is optimal for the GS model for PUFs. This is further expanded on in later sections of the paper.

The remainder of this paper is organized as follows. Section II provides the system model for PUF and formally defines the problem. Section III shows the algebraic binning method and polar code construction achieving the maximal key generation rate. Section IV compares our method to the existing other methods. Section V presents the simulation result. Finally, Section VI concludes the paper and gives future directions.

II. SYSTEM MODEL

Upper case letters represent random variables and lower case letters their realizations. A superscript denotes a vector of random variables, e.g., $X^n = X_1, X_2, \ldots, X_n$, and a subscript denotes the position of a variable in a vector. Calligraphic letters such as $\mathcal{X}$ denote sets, and set sizes are written as $|\mathcal{X}|$. $H_b(x) = -x \log x - (1-x) \log(1-x)$ is the binary entropy function. The $*$-operator is defined as $p \ast x = p(1-x) + (1-p)x$. The operator $\oplus$ represents the element-wise modulo-2 summation. A binary symmetric channel (BSC) with crossover probability $p$ is denoted by BSC$(p)$, $Q_b(X)$ represents a binary quantizer that quantizes $X > 0$ to 1 and $X < 0$ to 0.

A physically unclonable function (PUF) can be mathematically represented fairly simply as

$$Y = X + Z,$$
where \( Y \) is the PUF output, \( X \sim \mathcal{N}(0, P) \) is the static randomness and \( Z \sim \mathcal{N}(0, N) \) the dynamic randomness independent of \( X \). As stated earlier in the introduction, \( X \) is the desired “signal”, which is corrupted by the noise “\( Z \)” when observed at the output of a PUF.

In most conventional systems today, the PUF output is quantized immediately after observation. Most models in literature assume the output passes through a binary quantizer \( Q_{b}() \). It can be easily shown that \( Q_{b}(Y) \), \( Q_{b}(X) \) are distributed as Bernoulli(1/2), and \( Q_{b}(Y) = Q_{b}(X) + Z' \), where \( Z' \) is independent of \( Q_{b}(Y) \) and \( Q_{b}(X) \), distributed as Bernoulli(p), where \( p \) is a function of \( P \) and \( N \).

As mentioned earlier, we follow the generated-secret (GS) model for key generation based on PUFs, as depicted in Figure 1. For a given sequence \( X^{n} \), our task is to design an encoder \( \phi : \mathbb{F}_{2}^{n} \rightarrow (\mathbb{F}_{2}^{n-k}, \mathbb{F}_{2}^{k}) \) that generates a helper sequence \( S^{n-k} \) and a key \( W^{k} \) and a decoder \( \psi : \mathbb{F}_{2}^{n} \times \mathbb{F}_{2}^{k} \rightarrow \mathbb{F}_{2}^{k} \) that authenticates the key. This is such that, for a particular PUF, the probability of successful authentication goes to 1 as \( n \) goes to infinity. Let \((S^{n-k}, W^{k}) = \phi(X^{n}) \) and \( \hat{W}^{k} = \psi(Y^{n}, S^{n-k}) \).

\[
\Pr(\hat{W}^{k} \neq W^{k}) \rightarrow 0 \quad \text{as} \quad n \rightarrow \infty. \tag{1}
\]

Define the key generation rate as \( R = k/n \), we desire to determine the maximal key generation rate

\[
\max_{\phi, \psi} \quad R \\
\text{s.t.} \quad \Pr(\hat{W}^{k} \neq W^{k}) \rightarrow 0.
\]

### III. PUF System Design Using Polar Codes

In this section, we first state our main theorem and show the optimal key generation rate is achievable with algebraic binning using linear codes. Note that, the result can also be obtained by random binning, but algebraic binning offers greater insights for PUF key generation system design. Therefore, we choose to use an algebraic binning framework going forward.

#### A. Algebraic Binning Using Linear Codes

**Theorem 1.** Given a PUF and an associated key generation rate \( R < I(X; Y) \), there exists a linear code such that

\[
\Pr(\hat{W}^{k} \neq W^{k}) \rightarrow 0 \quad \text{as} \quad n \rightarrow \infty. \tag{2}
\]

\[
\text{Encoder}
\]

\[
\text{Helper Data } S^{n-k}
\]

\[
\text{Helper}
\]

\[
\text{PUF}
\]

\[
\text{Enrollment}
\]

\[
X^{n}
\]

\[
\text{Authentication}
\]

\[
Y^{n}
\]

\[
\text{Decoder}
\]

\[
\text{Key } \hat{W}^{i}
\]

\[
\text{Fig. 1. The PUF system model}
\]

\[
\text{Encoder}
\]

\[
\text{Helper Data } S^{n-k}
\]

\[
\text{Helper}
\]

\[
\text{H}
\]

\[
\text{S}^{n-k}
\]

\[
\text{Decoder}
\]

\[
\mod C
\]

\[
\text{Key } \hat{W}^{i}
\]

\[
\text{Fig. 2. The PUF system model with algebraic binning}
\]

**Proof.** The basic idea is to generate the bins as the cosets of a “good” parity-check code. Let an \((n, k)\) binary parity check code specified by \((n-k) \times n\) (binary) parity-check matrix \( H \). The code \( \mathcal{C} = \{c^{n}\} \) contains all \( n \)-length binary vectors \( c^{n} \) whose syndrome \( s^{n-k} = Hc^{n} \) is equal to zero, where here multiplication and addition are modulo 2. Assuming that all rows of \( H \) are linearly independent, there are \( 2^{k} \) codewords in \( \mathcal{C} \), so the code rate is \((\log |\mathcal{C}|)/n = k/n \). Given some general syndrome \( s^{n-k} \in \{0,1\}^{n-k} \), the set of all \( n \)-length vectors \( x^{n} \) satisfying \( Hx^{n} = s^{n-k} \) is called a coset \( \mathcal{C}_{s} \). Define a decoding function \( f(s^{n-k}) \), where \( f : \{0,1\}^{n-k} \rightarrow \{0,1\}^{n} \), is equal to the vector \( v^{n} \in \mathcal{C}_{s} \) with the minimum Hamming weight, where ties are broken evenly. It follows from linearity that the coset is a shift of the code \( \mathcal{C} \) by the vector \( v^{n} \), i.e.,

\[
\mathcal{C}_{s} \triangleq \{x^{n} : Hx^{n} = s^{n-k}\} = \{c^{n} \oplus v^{n} : c^{n} \in \mathcal{C}\} \triangleq \mathcal{C}^{v}
\]

where the \( n \)-vector \( v^{n} = f(s^{n-k}) \) is the coset leader.

Decoding of this parity-check code amounts to quantizing \( y^{n} \) to the nearest vector in \( \mathcal{C}^{v} \) with respect to the Hamming distance. This vector, \( \hat{y}^{n} \), can be computed by syndrome decoding using the function \( f(Hy^{n}) \).

\[
f(Hy^{n}) \text{ is an estimate of the noise } z^{n}. \text{ Alternatively, we can interpret } f(Hy^{n}) \text{ as the error vector in quantizing } y^{n} \text{ by } \mathcal{C},
\]

\[
f(Hy^{n}) = y^{n} \mod \mathcal{C} \tag{3}
\]

We may view the decoder above as a partition of \( \{0,1\}^{n} \) to \( 2^{k} \) decision cells of size \( 2^{n-k} \) each, which are all shifted versions of the basic “Voronoi” set

\[
\{z^{n} : z^{n} \oplus f(Hz^{n}) = 0\} \triangleq \Omega_{0}
\]

Each of the \( 2^{n-k} \) members of \( \Omega_{0} \) is a coset leader for a different coset.

We are interested in “good” parity check codes over binary symmetric channels (BSC) with crossover probability \( p \) that are capacity achieving, i.e., they have a rate \( R \) arbitrarily close
to $1 - H(p)$ for $n$ large enough. Now, choosing $p < 1/2$ such that
\[ H(p) = H(X|Y) \]

or together with Algorithm 1 will grant us the desired result.

**Algorithm 1**

**Algebraic Binning**

**procedure** ENROLLMENT($x^n$)

Store the syndrome $s^{n-k} = H x^n$ as helper data.

**end procedure**

**procedure** AUTHENTICATION($y^n, s^{n-k}$)

Find the coset leader $v^n = f(s^{n-k})$.

Find the corrector $y^n = (v^n + s^n) \mod C$.

Reconstruct the key $\hat{w}^k = y^n \oplus \hat{z}^n \oplus v^n$.

**end procedure**

Note that the decoding procedure in Algorithm 1 is unique, so unlike in random binning we never have ambiguous decoding. Hence, letting $z^n = x^n \oplus y^n$ and noting from (4) that $z^n = f(H(x^n \oplus y^n)) = f(Hz^n)$, a decoding error event amounts to $\{\hat{w}^k \neq w^k\} \leftrightarrow \{\hat{z}^n \neq z^n\}$ so the probability of decoding error is
\[ \Pr\{W^k \neq \hat{W}^k\} = \Pr\{X^n \neq \hat{X}^n\} = \Pr\{f(HZ^n) \neq Z^n\} \]

which by good BSC-$p$ code is smaller than $\epsilon$.

Because the total number of typical sequences are $2^{nH(X)}$, maximizing the key generation rate $R$ is equivalent to minimize the number of bins (cosets)
\[ \max R = \max \frac{k}{n} = \max 1 - \frac{n-k}{n} \quad (4) \]

The optimality is guaranteed by Slepian-Wolf bound for distributed source coding.

**Theorem 2** (Slepian-Wolf [5]). For the distributed source coding problem for the source $(X,Y)$ drawn i.i.d. $\sim p(x,y)$, the achievable rate region is given by
\[ R_1 \geq H(X|Y), \quad R_2 \geq H(Y|X), \quad R_1 + R_2 \geq H(X,Y). \]

To establish the connection between GS model and Slepian-Wolf problem, we see the two PUF responses $X$ and $Y$ are the correlated sources for Slepian-Wolf problem, and the number of bins in GS model is equivalent to the rate of the first source in Slepian-Wolf problem
\[ \frac{n-k}{n} = \frac{R_1}{H(X|Y)}. \quad (5) \]

Combining Equat. (4) and (5), we have
\[ \max R = 1 - \min \frac{n-k}{n} = 1 - H(X|Y) = H(X) - H(X|Y) = I(X;Y). \]

**Remark 1.** The proof showed the key generation rate $I(X;Y)$ is achievable with “good” parity check codes, in a sense that every coset is a good channel code over BSC($p$). It is a general statement, but as long as one can find the “good” parity check codes with each coset a good channel code for some channel, the key generation rate $I(X;Y)$ is achievable for that channel.

**B. Polar Codes for PUFs**

Polar codes are popular linear block codes, introduced by Arikan in [7]. A binary polar code can be specified by $(N,K,F,u^F)$, where $N = 2^n$ is the block length, $K$ is the number of information bits encoded per codeword, $F$ is the set of indices of the $N - K$ frozen bits and $u^F$ is a vector of frozen bits, which is known to both encoder and decoder.

1) **Encoding of Polar Codes:** For an $(N,K,F,u^F)$ polar code, the encoding operation for a message vector $u^N$, is performed using a generator matrix,
\[ G_N = G_{N2^{\otimes \log N}}, \]

where $B_N$ is a bit-reversal permutation matrix, $G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ and $\otimes$ denotes the Kronecker product.

Given a message vector $u^N$, the codewords are generated as
\[ x^n = u^n \oplus u^F(G_N)F, \]

where $F^c \triangleq \{1,2,\ldots,N\} \setminus F$ corresponds to the information bits indices. So $u^F$ are the information bits and $u^F$ are the frozen bits.

2) **Decoding of Polar Codes:** Polar codes achieve the channel capacity asymptotically in code length, when decoding is done using the successive-cancellation (SC) decoding algorithm. The SC decoder observes $(y^N,u^F)$ and generates an estimate $\hat{u}^N$ of $u^N$. The $i$th bit of the estimate $\hat{u}^N$ depends on the channel output $y^n$ and the previous bit decisions $\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_{i-1}$, denoted by $\hat{u}^{i-1}$. It uses the following decision rules,
\[ \hat{u}_i = \begin{cases} u_i & \text{if } i \in F \\ 0 & \text{if } i \in F^c \text{ and } L_i(y^N, \hat{u}^{i-1}) \geq 1 \\ l & \text{if } i \in F^c \text{ and } L_i(y^N, \hat{u}^{i-1}) < 1 \end{cases} \]

where $L_i(y^N, \hat{u}^{i-1}) = P(y^N, \hat{u}^{i-1}|0)/P(y^N, \hat{u}^{i-1}|1)$ is the $i$th likelihood ratio (LR) at length $N$. We omit further details in SC decoding for limited space, readers can get the full knowledge of SC decoding in [7].

**C. Applying Polar Codes to PUFs**

Given the noise of PUF as a BSC($p$), block length $N$ and rate $K/N$, we have the polar code with parameters $(N,K,F)$. And the algebraic binning with polar code is shown in Algorithm 2.

**Theorem 3.** For PUF, every key generation rate $R < I(X;Y)$, there exist a polar encoder and decoder, such that
\[ \sum_{s_{N-K} \in X^{N-K}} \frac{1}{2^{N-K}} \sum_{u_K \in \hat{u}_K} \frac{1}{2^K} \Pr(\hat{w}^K = \hat{w}^K) = O(N^{-1/4}). \]

□
Algorithm 2 Algebraic Binning with Polar Code Construction

procedure ENROLLMENT($x^N$, $(N, K, F)$)
    Store the syndrome $s^{N-K} = ((G^T_N)^{-1}x^N)_F$ as helper data.
end procedure

procedure AUTHENTICATION($y^N$, $s^{N-K}$, $(N, K, F)$)
    Reconstruct the key $\hat{w}^K = SC_{dec}(y^N, s^{N-K})$.
end procedure

Proof. As introduced in [7], polar code can be represented as

$$x^N = u^N G_N = u^{F_n}(G_N)_F \oplus u^{F}(G_N)_F,$$

in (a) we abuse the notations $u^K$ and $s^{N-K}$ since $u^{F_n}$ and $u^{F}$ are length $N$ vector although they have all zeros on the indices out of $F_n$ and $F$ respectively. By inversion of $G_N$, we have the syndrome and the key as

$$s^{N-K} = ((G^T_N)^{-1}x^N)_F,$$

$$w^K = ((G^T_N)^{-1}x^N)_F.$$

Now for each PUF observation $x^N$, we treat it as a codeword of polar code with parameters $(N, K, F, s^{N-K})$. So we use a set of polar code $C = \{C(N, K, F, s^{N-K}) : s^{N-K} \in \mathbb{F}^{N-K}_2\}$. Because $G_N$ has full rank, for any $x^N \in \mathbb{F}^{N}_2$, $x^N \in C$, and $C \subseteq \mathbb{F}^{N}_2$. So $C = \mathbb{F}^{N}_2$. We proved that $C$ is a coset partition of $\mathbb{F}^{N}_2$ and each coset code $C(N, K, F, s^{N-K})$ with coset leader $s^{N-K}$ is a channel code for the channel. According to Theorem 3 in [7], we have for rate $R < I(X;Y)$, the block error probability for polar coding under successive cancellation decoding satisfies

$$\sum_{s^{N-K} \in \mathbb{F}^{N-K}_2} \frac{1}{2^{N-K}} \sum_{u^K \in \mathbb{F}^{K}_2} \frac{1}{2^K} \Pr(\hat{w}^K \neq w^K) = O(N^{-\frac{1}{2}}).$$

Although polar codes cannot guarantee each coset code is a good channel code such that

$$\Pr(\hat{W}^K \neq W^K) = o(1),$$

on average, we obtain a good coset partition as required by Theorem 1.

D. Achievable Scheme for Unquantized PUFs: The Gaussian Case

As mentioned earlier, a vast majority of PUF outputs are quantized to a binary alphabet right after generation. However, for the case when $X^n$ and $Z^n$ remain as Gaussian sequences, we use a lattice based coding scheme as below.

Definitions: Lattice $\Lambda$ is a discrete subgroup of $\mathbb{R}^n$. Quantization with respect to $\Lambda$ is $Q_{\Lambda}(x^n) = \arg \min_{\lambda \in \Lambda} \|x^n - \lambda\|$. Fundamental Voronoi region of $\Lambda$ is $V(\Lambda) = \{x^n : Q_{\Lambda}(x^n) = 0\}$. Volume of the Voronoi region of $\Lambda$ is $V(\Lambda) = \int_{V(\Lambda)} dx^n$. Normalized second moment of $\Lambda$ is $G(\Lambda) = \frac{\sigma^2(\Lambda)}{V(\Lambda)^{\frac{n}{2}}}$ where $\sigma^2(\Lambda) = \frac{1}{V(\Lambda)^2} \int_{V(\Lambda)} \|x^n\|^2 dx^n$. A pair of Lattices $(\Lambda, \Lambda_0)$ are said to be nested if $\Lambda \subseteq \Lambda_0$.

We used nested lattices $\Lambda \subseteq \Lambda_0$ for coset partitioning and algebraic binning. The encoder block with input $x^n$ and output $d^n$ is implemented by lattice modulo operation

$$d^n = [x^n] \mod \Lambda_0 = x^n - Q_{\Lambda_0}(x^n).$$

We use $d^n$ as a helper data. For the decoder block with input $y^n$, helper data $d^n$, and output $u^n$, we perform

$$u^n = Q_{\Lambda_0}([y^n - d^n] \mod \Lambda).$$

Since

$$y^n = x^n + z^n = t^n + d^n + z^n$$

where $t^n = Q_{\Lambda_0}(x^n) \in \Lambda_0$, the decoder output is

$$u^n = Q_{\Lambda_0}([t^n + z^n] \mod \Lambda).$$

If we use nested lattices satisfying $z^n \in V(\Lambda_0)$ with high probability, it follows that

$$Q_{\Lambda_0}([t^n + z^n] \mod \Lambda) = Q_{\Lambda_0}(t^n)$$

with high probability. Since

$$Q_{\Lambda_0}(t^n) = t^n = Q_{\Lambda_0}(x^n),$$

it also means that

$$Q_{\Lambda_0}([y^n - d^n] \mod \Lambda) = Q_{\Lambda_0}(x^n)$$

with high probability. In other words, the helper data cancels the effect of noise $z^n$.

The lattice codebook is defined by the set $\Lambda_0 \cap V(\Lambda)$. The code rate is given by $R = \frac{1}{n} \log \left( \frac{V(\Lambda)}{V(\Lambda_0)} \right)$ where $V(\cdot)$ is the volume of the fundamental Voronoi region of a lattice. We use nested lattices with parameters $\sigma^2(\Lambda) = P$, $G(\Lambda) = \frac{1}{2^{\frac{n}{2}}}P$, and $V(\Lambda) = \sqrt{2\pi e} P^{\frac{n}{2}}$. Nested lattices good for Gaussian channel coding [14] can be used to achieve a rate up to $R = \frac{1}{n} \log \left( \frac{P}{2\pi e} \right)$ with vanishing error probability. In practice, polar lattices [12], [13] can be used for polynomial-time processing.

IV. COMPARISONS WITH EXISTING METHODS

There are several existing method proposed for the GS model.

The authors et al. [9] established the connection between Wyner-Ziv problem and the GS model, and described the key-leakage-storage region for GS model. However, under GS model, according to the definitions, storage rate and privacy rate are the same since $I(X^n;W) = H(W) - H(W|X^n) = H(W)$, where $W$ is a function of $X^n$ in GS model. It is also reflected in Theorem 1 in [9] as $R_l$ and $R_w$ have the same bound. So the key-leakage-privacy can be treated as key-leakage region or key-storage region, and we describe the optimal point of the key-storage region by the algebraic binning argument. The authors et al. [9] also show a polar code construction based on the nested polar code in [10] to achieve the key-leakage-storage region, which give the optimal key generation rate as $1 - H_q(q*p)$ for given PUF noise as a BSC(p), where $q \in [0, 0.5]$ is a chosen parameter for the
first step vector quantization (VQ) in the nested polar code. Since \( H_b(q \ast p) \geq H_b(p) \), we have our optimal rate greater than their optimal rate \( 1 - H_b(p) \geq 1 - H_b(q \ast p) \), and the storage \( H_b(p) \leq H_b(q \ast p) \). Notice that the both equalities can be achieved if \( q = 0 \), but at this point they will lose the nested polar code construction. The reason for this is the gap between Wyner-Ziv problem having distortion (reflected as the first step VQ in the nested polar code construction) and the GS model requires an exact recovery of the key. So the VQ step introducing the distortion is not necessary for GS model. In all, we offer a better rate with a simpler implementation.

The authors et al. [11] offer a LDPC based scheme for PUF. But it does not optimize the key generation rate since the LDPC does not necessarily form a coset partition.

V. SIMULATION RESULTS

We simulate the system in Figure 2 with the polar code construction in Section III-C in MATLAB. If we use PUFs in a field programmable gate array (FPGA) as the randomness source, we must satisfy a block error probability \( P_B \) of at most \( 10^{-6} \) [16]. Consider a BSC\((p)\) with crossover probability \( p = 0.15 \), which is a common value for SRAM PUFs.

First, we consider the block length \( N = 1024 \) and we design polar code with rate \( K/N = 128/1024 = 0.125 \) for the BSC\((p)\) channel. We evaluate the block error performance of this code with SC decoder and SC list (SCL) decoder with list size \( 8 \) respectively for a BSC with a range of crossover probability, as shown in Figure 3. It shows the SCL decoder has better performance, and achieves a block error probability of \( P_B = 10^{-6} \) at a crossover probability \( 0.2 \). For comparison, we achieve the key generation rate 0.125 with crossover probability and block error probability \((0.2, 10^{-6})\), better than the crossover probability and block error probability tuple \((0.1819, 10^{-6})\) in [9].

![Fig. 3. Block error probability over a BSC\((p)\) with SC decoder and SCL decoder with block length 1024, respectively.](image)

VI. CONCLUSION

The achievability of optimal key generation rate can be proved by random binning with typicality argument, but this point of view does not offer the opportunity to explore the usage of structures of codes. We show what kind of code people need for the design of PUF system to achieve the optimal key generation rate by algebraic binning. We design a polar code-based system for PUFs that achieve better key generation rate than existing methods. In future work, we will study the “good” code for unquantized PUFs.

ACKNOWLEDGMENT

This work was supported by the NSF and the ONR.

REFERENCES

[1] Gassend, Blaise, Dwayne Clarke, Marten Van Dijk, and Srinivas Devadas. “Silicon physical random functions.” In Proceedings of the 9th ACM conference on Computer and communications security, pp. 148-160. ACM, 2002.
[2] Suh, G. Edward, and Srinivas Devadas. “Physical unclonable functions for device authentication and secret key generation.” In Proceedings of the 44th annual Design Automation Conference, pp. 9-14. ACM, 2007.
[3] Guajardo, Jorge, Sandeep S. Kumar, Geert-Jan Schrijen, and Pim Tuyls. “FPGA intrinsic PUFs and their use for IP protection.” In International workshop on Cryptographic Hardware and Embedded Systems, pp. 63-80. Springer Berlin Heidelberg, 2007.
[4] Pelgrom, Marcel JM, Aad CJ Duinmaijer, and Anton PG Welbers. “Matching properties of MOS transistors.” IEEE Journal of solid-state circuits 24, no. 5 (1989): 1433-1439.
[5] Cover, Thomas M., and Joy A. Thomas. Elements of information theory. John Wiley & Sons, 2012.
[6] Ignatenko, Tanya, and Frans MJ Willems. “Biometric systems: Privacy and secrecy aspects.” IEEE Transactions on Information Forensics and Security 4.4 (2009): 956-973.
[7] Arikan, Erdal. “Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels.” IEEE Transactions on Information Theory 55.7 (2009): 3051-3073.
[8] B. Chen, T. Ignatenko, F. M. Willems, R. Maes, E. van der Sluis, and G. Selznaes, “A robust SRAM-PUF key generation scheme based on polar codes,” July 2017, [Online]. Available: arxiv.org/abs/1707.07320.
[9] Onur Gunlu, Onurcan Iscan, Vladimir Sidorenko, and Gerhard Kramer, “Wyner-Ziv Coding for Physical Unclonable Functions and Biometric Secrecy Systems,” Sep 2017, [Online]. Available: https://arxiv.org/abs/1709.00275.
[10] Korada, Satoshi Babu, and Rudiger Urbanke. “Polar codes for sleipn-wolf, wyner-ziv, and gelfand-pinsker.” Information Theory (ITW 2010, Cairo), 2010 IEEE Information Theory Workshop on. IEEE, 2010.
[11] Muech, Sven, and Martin Bossert. “A New Error Correction Scheme for Physical Unclonable Functions.” SCC 2017; 11th International ITG Conference on Systems, Communications and Coding; Proceedings of. VDE, 2017.
[12] Y. Yan, C. Ling, and X. Wu, “Polar lattices: Where Arikan meets Forney,” Proc. IEEE Int. Symp. Inform. Theory, Istanbul, Jul. 2013.
[13] Y. Yan, L. Liu, and C. Ling, “Polar lattices for strong secrecy over the mod-A Gaussian wiretap channel,” Jan. 2014. [Online] Available: arXiv:1401.4532 [cs.IT].
[14] U. Erez and R. Zamir, “Achieving \( \frac{1}{2} \log(1 + \text{SNR}) \) on the AWGN channel with lattice encoding and decoding,” IEEE Trans. Inf. Theory, vol. 50, no. 10, pp. 2293–2314, Oct. 2004.
[15] Wyner, Aaron, and Jacob Ziv. “The rate-distortion function for source coding with side information at the decoder.” IEEE Transactions on Information Theory 22.1 (1976): 1-10.
[16] Bosch, Christoph, et al. “Efficient helper data key extractor on FPGAs.” Cryptographic Hardware and Embedded Systems”CHES 2008 (2008): 181-197.