Wyner-Ziv Coding for Physical Unclonable Functions and Biometric Secrecy Systems

Onur Günlü, *Student Member, IEEE,* Onurcan İşcan, Vladimir Sidorenko, *Member, IEEE,* and Gerhard Kramer, *Fellow, IEEE*

***Abstract*—Two constructive linear coding methods, previously proposed for the Wyner-Ziv problem, are developed to achieve all points of the key-leakage-storage regions of the generated- and chosen-secret models of biometric secrecy systems. The models also apply to physical unclonable functions that are realized by fine hardware variations. The linear coding is implemented by using nested polar codes and the designs achieve privacy-leakage and storage rates that improve on existing methods.**

***Index Terms*—Information theoretic privacy, secret key agreement, physical unclonable functions, Wyner-Ziv coding.**

I. Introduction

Biometric features like fingerprints are unique and can be used to authenticate and identify individuals, and to generate secret keys. Similarly, one can generate secret keys with physical unclonable functions (PUFs) that are used as sources of randomness. For example, fine variations of ring oscillator outputs and the start-up behavior of static random access memories (SRAM) can serve as PUFs [1]. Fingerprints and PUFs are identifiers with high entropy and reliable outputs [2], [3], and one can consider them as physical "one-way functions" that are easy to compute and difficult to invert [4].

Various applications use the keys generated from biometric or physical identifier outputs. For instance, a fifth generation (5G) mobile device can use its SRAMs as a source to extract secret keys to encrypt data. Similarly, digital internet-of-things (IoT) devices that carry sensitive data, e.g., wearable or e-health devices, can use a physical identifier to store secret keys [5] so that only a mobile device with access to this secret key can control the IoT device. This application provides low-complexity data privacy; see, for instance, [6] for other applications of PUFs in 5G wireless networks.

PUFs provide security against invasive attacks since their outputs permanently change if an attacker tampers with the device [3]. Physical-layer security solutions proposed for wireless networks do not generally provide security against such attacks. Therefore, *hardware-intrinsic security* for 5G wireless networks provides a low-complexity security and privacy solution that is complementary to physical-layer security solutions.

Replacing biometric and physical identifiers is often expensive, or not possible, and this means that the same identifier is used many times. One should, therefore, find the limits for the information leaked to an eavesdropper about the identifier (*privacy leakage*) and about the secret key (*secrecy leakage*) as well as the limits for the amount of storage. We consider two models for leakage and storage analysis called the *generated-secret (GS)* and the *chosen-secret (CS) models*.

A. Related Work

We briefly review the past work. The region of achievable secret-key vs. privacy-leakage (key vs. leakage) rates for the GS and CS models are given in [2] and [7]. The storage rates for different secrecy-leakage rates and multiple encoder measurements are analyzed, respectively, in [8] and [9]. The binary Golay code is used in [2] as a vector quantizer in combination with Slepian-Wolf (SW) coding techniques [10] to illustrate that the key vs. leakage rate ratio can be increased via quantization.

A polar code based construction is proposed in [5] for the GS model. This construction assumes a "private" key shared only by an encoder and a decoder, which is not realistic since a private key requires hardware protection against invasive attacks. If such a protection is possible, then there is no need to use an on-demand key storage method like a PUF.

B. Motivation and Organization

The bounds on the storage rate of the GS model and on the Wyner-Ziv (WZ) rate have the same expressions. We further show that there is a code that satisfies the constraints of both problems, corresponding to a functional equivalence of the two problems. Extending this equivalence, we propose code constructions that asymptotically achieve all points of the key-leakage-storage region, and that improve on existing methods in [5], [11], and [12]. We consider both key generation (GS model) and embedding (CS model) problems to address different practical scenarios.

This paper is organized as follows. In Sections II-A and II-B, we describe the GS and CS models, the WZ problem, and give their rate regions. In Section II-C, we prove that there is a code that satisfies the constraints of the WZ problem and the GS model simultaneously to motivate using WZ coding techniques for key generation and embedding problems. Section III describes a constructive method and proves its

The work of O. Günlü was supported by the German Research Foundation (DFG) through the project HoliPUF under the grant KR3517/6-1. V. Sidorenko is on leave from the Institute for Information Transmission Problems, Russian Academy of Science. The work of G. Kramer was supported by an Alexander von Humboldt Professorship endowed by the German Federal Ministry of Education and Research.

O. Günlü, V. Sidorenko, and G. Kramer are with the Institute for Communications Engineering, Technical University of Munich, Munich, Germany (e-mail: {onur.gunlu, vladimir.sidorenko, gerhard.kramer}@tum.de).

O. İşcan is with the Huawei European Research Center, Munich, Germany (email: onurcan.iscan@huawei.com).

asymptotic optimality for the GS and CS models. We improve existing methods, and we show their suboptimality even after known improvements in Section IV. The gain is in the privacy-leakage rates. We design nested polar codes in Section V. We use the WZ code construction proposed in [13] to generate secret keys from SRAM PUFs with a block-error probability of at most 10^{-6} and a minimum secret-key size of 128 bits. The proposed codes achieve operating points that cannot be achieved by existing methods.

C. Notation

Upper case letters represent random variables and lower case letters their realizations. A superscript denotes a string of variables, e.g., $X^n = X_1 \dots X_i \dots X_n$, and a subscript denotes the position of a variable in a string. A random variable X has probability distribution P_X. Calligraphic letters such as $\mathcal{X}$ denote sets, and set sizes are written as $|\mathcal{X}|$. Bold letters such as $\mathbf{H}$ represent matrices. $\mathcal{T}_\epsilon^n(P_X)$ denotes the set of length-n letter-typical sequences with respect to the probability distribution P_X and the positive number ϵ [14]. $\mathsf{Enc}(\cdot)$ is an encoder mapping and $\mathsf{Dec}(\cdot)$ is a decoder mapping. $X-Y-Z$ indicates a Markov chain. $H_b(x) = -x\log x - (1-x)\log(1-x)$ is the binary entropy function. The $*$-operator is defined as $p * x = p(1-x) + (1-p)x$. The operator $\oplus$ represents the element-wise modulo-2 summation. A binary symmetric channel (BSC) with crossover probability p is denoted by BSC(p). $X^n \sim \text{Bern}^n(\alpha)$ denotes that X^n is an independent and identically distributed (i.i.d.) binary sequence of random variables with $\Pr[X_i = 1] = \alpha$ for $i = 1, 2, \dots, n$. The superscript T represents the transpose.

II. Problem Formulations

A. Generated- and Chosen-secret Models

Consider the GS model in Fig. 1(a), where a secret key is generated from a biometric or physical source. The source $\mathcal{X}$, measurement $\mathcal{Y}$, secret key $\mathcal{S}$, and storage $\mathcal{W}$ alphabets are finite sets. During enrollment, the encoder observes an i.i.d. sequence X^n, generated by the identifier (source) according to some P_X, and generates a secret key S and public helper data W as $(S, W) = \mathsf{Enc}(X^n)$. During reconstruction, the decoder observes a noisy source measurement Y^n of X^n through a memoryless channel $P_{Y|X}$ together with the helper data W. The decoder estimates the secret key as $\widehat{S} = \mathsf{Dec}(Y^n, W)$. We remark that we assume that an eavesdropper cannot observe a sequence correlated with the identifier outputs because, for many physical identifiers and some biometric identifiers, an invasive attack that obtains a correlated sequence permanently changes the identifier output.

Fig. 1(b) shows the CS model, where a secret key $S' \in \mathcal{S}$ is embedded into the helper data as $W' = \mathsf{Enc}(X^n, S')$. The decoder for the CS model estimates the secret key as $\widehat{S}' = \mathsf{Dec}(Y^n, W')$.

Definition 1. A key-leakage-storage tuple (R_s, R_l, R_w) is *achievable* for the GS or CS models if, given any $\delta > 0$, there is some $n \geq 1$, an encoder, and a decoder such that

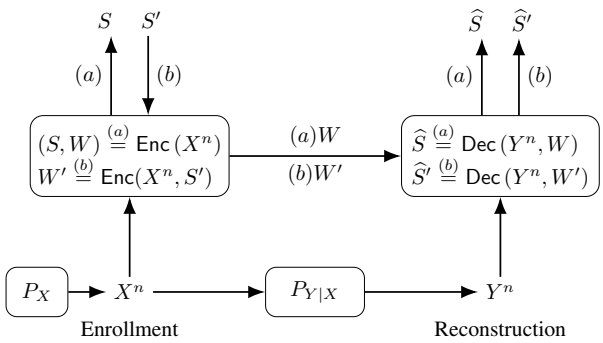

Fig. 1. The (a) GS and (b) CS models.

$R_s = \frac{\log|\mathcal{S}|}{n}$ and

$$\Pr[\widehat{S} \neq S] \leq \delta \qquad (reliability) \tag{1}$$

$$\frac{1}{n} I(S; W) \leq \delta \qquad (secrecy) \tag{2}$$

$$\frac{1}{n} H(S) \geq R_s - \delta \qquad (uniformity) \tag{3}$$

$$\frac{1}{n} \log|\mathcal{W}| \leq R_w + \delta \qquad (storage) \tag{4}$$

$$\frac{1}{n} I(X^n; W) \leq R_l + \delta. \qquad (privacy) \tag{5}$$

The *key-leakage-storage* regions $\mathcal{R}_{\text{gs}}$ and $\mathcal{R}_{\text{cs}}$ for the GS and CS models, respectively, are the closures of the sets of achievable tuples for the corresponding models. $\lozenge$

Theorem 1 (Ignatenko and Willems [2])**.** *The key-leakage-storage regions for the GS and CS models, respectively, are*

$$\begin{aligned}\mathcal{R}_{gs} = \bigcup_{P_{U|X}} \Big\{ & (R_s, R_l, R_w) : 0 \leq R_s \leq I(U;Y), \\ & R_l \geq I(U;X) - I(U;Y), \\ & R_w \geq I(U;X) - I(U;Y) \ \textit{for} \\ & P_{UXY} = P_{U|X} P_X P_{Y|X} \Big\},\end{aligned} \tag{6}$$

$$\begin{aligned}\mathcal{R}_{cs} = \bigcup_{P_{U|X}} \Big\{ & (R_s, R_l, R_w) : 0 \leq R_s \leq I(U;Y), \\ & R_l \geq I(U;X) - I(U;Y), \\ & R_w \geq I(U;X) \ \textit{for} \\ & P_{UXY} = P_{U|X} P_X P_{Y|X} \Big\}.\end{aligned} \tag{7}$$

The alphabet $\mathcal{U}$ of the auxiliary random variable U can be limited to have size $|\mathcal{U}| \leq |\mathcal{X}| + 1$ for both regions.

We give a sketch of the proof to motivate our code construction in the next section.

Proof: Fix a $P_{U|X}$. Randomly and independently generate codewords $u^n(w, s)$, $w = 1, 2, \dots, 2^{nR_w}$, $s = 1, 2, \dots, 2^{nR_s}$ according to $\prod_{i=1}^n P_U(u_i)$, where $P_U(u_i) = \sum_{x \in \mathcal{X}} P_{U|X}(u|x) P_X(x)$. These codewords define the random codebook

$$\tilde{\mathcal{C}} = \{U^n(w, s)\}_{(w,s)=(1,1)}^{(2^{nR_w}, 2^{nR_s})}. \tag{8}$$

Let $0 < \epsilon' < \epsilon$. We first consider the GS model.

Encoding: Given x^n, the encoder looks for a codeword that is jointly typical with x^n, i.e., $(u^n(w,s),x^n)\in\mathcal{T}_{\epsilon'}^n(P_{UX})$. If there is one or more such codeword, the encoder chooses one of them and puts out (w,s). If there is no such codeword, set $w=s=1$. The encoder publicly stores the label w.

Decoding: The decoder puts out $\hat{s}$ if there is a unique key label $\hat{s}$ that satisfies the typicality check $(u^n(w,\hat{s}),y^n)\in\mathcal{T}_{\epsilon}^n(P_{UY})$; otherwise, it sets $\hat{s}=1$.

Using standard arguments, there is a code that satisfies (1)-(5) if we consider large n and approximately $2^{n(I(U;X)-I(U;Y))}$ storage labels w and $2^{nI(U;Y)}$ key labels s, i.e., $R_w\approx I(U;X)-I(U;Y)$ and $R_s\approx I(U;Y)$. The converse follows from standard properties of the entropy function.

We now consider the CS model where we embed a secret key s'. We add a one-time pad step to the encoding for the GS model and store the new helper data $w'=(w,s\oplus s')$. The decoder of the GS model puts out the estimate $\hat{s}$ so that one can obtain the new key estimate for the CS model as $\hat{s}'=\hat{s}\oplus(s\oplus s')$. Since the error probability $\Pr[\hat{S}'\neq S']$ is the same as the error probability $\Pr[\hat{S}\neq S]$ for the GS model, we obtain the same secret-key and privacy-leakage rates. Due to the additional helper data $s\oplus s'$, the storage rate is the sum of the secret-key and storage rates of the GS model. The converse follows from standard arguments. ■

B. Wyner-Ziv Problem

Consider two dependent random variables X and Y jointly generated according to P_{XY}. Fig. 2 depicts the WZ problem. The source $\mathcal{X}$, $\mathcal{Y}$, and side information $\mathcal{W}$ alphabets are finite sets. An encoder that observes X^n generates side information $W\in[1,2^{nR_w}]$. The decoder observes Y^n and W and puts out a quantized version $\widehat{X}^n$ of X^n. Define the average distortion between X^n and the reconstructed sequence $\widehat{X}^n$ as

$$\frac{1}{n}\sum_{i=1}^{n}E[d(X_i,\widehat{X}_i(Y^n,W))]$$

where $d(x,\hat{x})$ is a distortion function and $\hat{x}(y^n,w)$ is a mapping. For simplicity, assume that $d(x,\hat{x})$ is bounded.

Definition 2. A WZ rate-distortion pair (R_w,D) is *achievable* for a distortion measure $d(x,\hat{x})$ if, given any $\delta>0$, there is some $n\geq 1$, an encoder, and a decoder for which $R_w=\frac{\log|\mathcal{W}|}{n}$ and

$$\frac{1}{n}\sum_{i=1}^{n}E[d(X_i,\widehat{X}_i(Y^n,W))]\leq D+\delta. \tag{9}$$

The WZ rate-distortion region $\mathcal{R}_{\text{WZ}}$ is the closure of the set of achievable rate-distortion pairs. ◊

Theorem 2 (Wyner and Ziv [15])**.** *The WZ rate-distortion region is*

$$\mathcal{R}_{WZ}=\bigcup_{P_{U|X}}\bigcup_{f(U,Y)}\Big\{(R_w,D):R_w\geq I(U;X)-I(U;Y),\ D\geq E[d(X,f(U,Y))]\ \textit{for}\ P_{UXY}=P_{U|X}P_{XY}\Big\} \tag{10}$$

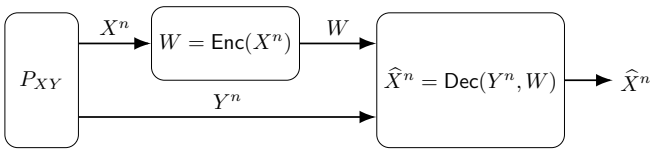

Fig. 2. The WZ problem.

where $f(U,Y)=\widehat{X}$ is a reconstruction function used at the decoder. One can limit the alphabet $\mathcal{U}$ of the auxiliary random variable U to have size $|\mathcal{U}|\leq|\mathcal{X}|+1$.

We give a sketch of the proof that we use later.

Proof: Fix a $P_{U|X}$ and $f(u,y)$ such that $E[d(X,\hat{X})]\leq D/(1+\epsilon)$ for some distortion D and $\epsilon>0$. Randomly and independently generate codewords $u^n(w,s)$, $w=1,2,\ldots,2^{nR_w}$, $s=1,2,\ldots,2^{nR_s}$ according to $\prod_{i=1}^n P_U(u_i)$. We use the same encoding and decoding steps as in the achievability proof of the GS model. After finding a $u^n(w,\hat{s})$ that satisfies $(u^n(w,\hat{s}),y^n)\in\mathcal{T}_{\epsilon}^n(P_{UY})$, the WZ decoder puts out $f(u_i,y_i)=\hat{x}_i$ for all $i=1,2,\ldots,n$.

Using standard arguments, there is a code that satisfies (9) if we consider large n and approximately $2^{n(I(U;X)-I(U;Y))}$ side information (storage) labels w, i.e., $R_w\approx I(U;X)-I(U;Y)$. We then have approximately $2^{nI(U;Y)}$ s labels. The converse follows from standard arguments. ■

C. Functional Equivalence

We first give an equivalence result.

Theorem 3. *Consider the GS model with the probability distributions P_X and $P_{Y|X}$, and the WZ problem with the joint probability distribution $P_{XY}=P_XP_{Y|X}$ and a distortion function $d(x,\hat{x})$. For every probability distribution $P_{U|X}$ and reconstruction function $f(\cdot)$, and a distortion D that satisfies the distortion constraint in (10), there is some $n\geq 1$, an encoder, and a decoder that satisfy (1)-(5) and (9) simultaneously.*

Proof: Consider the GS model with an additional distortion constraint as in (9). Fix some $P_{U|X}$ and $f(u,y)$ such that $|\mathcal{U}|\leq|\mathcal{X}|+1$ and $E[d(X,\hat{X})]\leq D/(1+\epsilon)$ for some distortion D and $\epsilon>0$. Use the code construction, encoder, and decoder given in the achievability proof of Theorem 1 so that one, asymptotically, achieves a key-leakage-storage (R_s,R_l,R_w) tuple. Using the typical average lemma [16, Section 2.4], the rate-distortion (R_w,D) pair can be achieved as well. ■

Based on Theorem 3, we consider the GS and WZ problems as *functionally equivalent*. Note that *functional duality* (see, e.g., [17]) is related to functional equivalence, but we do not exchange the encoders and decoders, unlike for functional duality. One can also show a functional duality of the GS model and the Gelfand-Pinsker problem [18], which is shown in [17] to be a functional dual of the WZ problem.

Motivated by Theorem 3 and using the WZ code construction proposed in [19] for linear codes, we show in the next section that such a construction with an additional privacy amplification step achieves the boundary points of the key-

leakage-storage regions of the GS and CS models for binary sources measured through a BSC.

III. Code Construction

Consider the lossy source coding method proposed in [19] that achieves the WZ rate-distortion function by using linear codes. We combine this approach with a privacy amplification step to achieve the boundary points of $\mathcal{R}_{\text{gs}}$ and $\mathcal{R}_{\text{cs}}$ for the GS and CS models, respectively, for a binary uniform identifier source and a BSC $P_{Y|X}$ with crossover probability p_A. We remark that there are algorithms to obtain almost i.i.d. uniform outputs from correlated and biased outputs (see, e.g., [1], [20]). Fig. 3(a) and Fig. 3(b) plot the WZ code constructions, respectively, for the GS and CS models.

Code Construction: Define the full-rank parity-check matrices $\mathbf{H}_1$, $\mathbf{H}_2$, and $\mathbf{H}$ as

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \end{bmatrix} \tag{11}$$

where $\mathbf{H}_1$ with dimensions $m_1 \times n$ defines a binary linear code $\mathcal{C}_1$ and $\mathbf{H}_2$ with dimensions $m_2 \times n$ defines another binary linear code $\mathcal{C}_2$. The code $\mathcal{C}$ defined by $\mathbf{H}$ in (11) is thus a subcode of $\mathcal{C}_1$. For some $q \in [0, 0.5]$, impose the conditions

$$\frac{m_1}{n} = H_b(q) + \delta \tag{12}$$

$$\frac{m_1 + m_2}{n} = H_b(q * p_A) + 2\delta \tag{13}$$

for some $\delta > 0$. First, apply vector quantization (VQ) to a uniformly distributed identifier output X^n by using $\mathcal{C}_1$. We then generate and store public helper data by using $\mathcal{C}_2$.

Enrollment: The VQ in Fig. 3 quantizes the source output X^n into the closest codeword X_q^n in $\mathcal{C}_1$ in Hamming metric. Define the error sequence

$$E_q^n = X^n \oplus X_q^n. \tag{14}$$

In the GS model, we publicly store the side information

$$W = X_q^n \mathbf{H}_2^T. \tag{15}$$

We then choose a deterministic function $g(\cdot)$ for privacy amplification [21] to generate a secret key as $S = g(X_q^n)$ such that

$$\begin{aligned} H(S) &= H(X_q^n) - H(X_q^n \mathbf{H}^T) \\ &\overset{(a)}{=} (n - m_1) - H(X_q^n \mathbf{H}^T) \end{aligned} \tag{16}$$

where (a) follows from the linearity of the code $\mathcal{C}_1$ and uniformity of X^n. Since X_q^n is uniformly distributed, the min-entropy and collision entropy are equal to the Shannon entropy and one can use universal families of hash functions or extractors for privacy amplification [21].

For the CS model shown in Fig. 3(b), we have access to an embedded (chosen) secret key S' that is independent of other random variables and such that $|\mathcal{S}| = |\mathcal{S}'|$. We store the helper data $W' = [W,\, S \oplus S']$.

Reconstruction: The noisy identifier output observed during reconstruction is $Y^n = X^n \oplus Z^n$, where Z^n is independent of X^n and $Z^n \sim \text{Bern}^n(p_A)$. The error sequence E_q^n and the noise sequence Z^n are independent. Furthermore, E_q^n asymptotically resembles an i.i.d. sequence $\sim \text{Bern}^n(q)$ when $n \to \infty$ [19], [22]. Therefore, when $n \to \infty$, the sequence $E_q^n \oplus Z^n$, which asymptotically corresponds to the noise sequence of the equivalent channel $P_{Y^n|X_q^n}$, is distributed according to $\text{Bern}^n(q * p_A)$. One can thus reconstruct X_q^n with high probability when $n \to \infty$ by using the syndrome decoder $f(\cdot)$ of the code $\mathcal{C}$ as follows

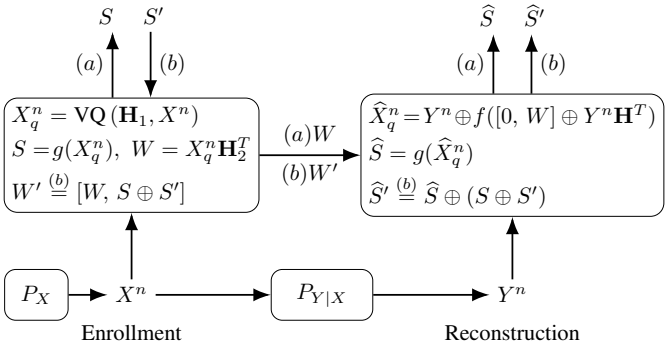

Fig. 3. WZ constructions for the (a) GS and (b) CS models.

$$\begin{aligned} \widehat{X}_q^n &= Y^n \oplus f([0,\, W] \oplus Y^n \mathbf{H}^T) \\ &\overset{(a)}{=} Y^n \oplus f(X_q^n \mathbf{H}^T \oplus Y^n \mathbf{H}^T) \\ &\overset{(b)}{=} (X_q^n \oplus E_q^n \oplus Z^n) \oplus f((E_q^n \oplus Z^n)\mathbf{H}^T) \\ &\overset{(c)}{=} (X_q^n \oplus E_q^n \oplus Z^n) \oplus (E_q^n \oplus Z^n) \\ &= X_q^n \end{aligned} \tag{17}$$

where (a) follows from (15) and because X_q^n is a codeword of $\mathcal{C}_1$, (b) follows from (14), and (c) follows with high probability because, asymptotically, $E_q^n \oplus Z^n \sim \text{Bern}^n(q * p_A)$ so that the syndrome decoder $f(\cdot)$ determines the noise sequence $E_q^n \oplus Z^n$ in the equivalent channel $P_{Y^n|X_q^n}$ with high probability.

The secret-key is reconstructed in the GS model as

$$\widehat{S} = g(\widehat{X}_q^n) \tag{18}$$

and in the CS model as

$$\widehat{S}' = g(\widehat{X}_q^n) \oplus (S \oplus S') \tag{19}$$

both of which result in the same error probability.

A. Optimality of GS Model Rates

Recall that $X^n \sim \text{Bern}^n(\frac{1}{2})$ and that the channel $P_{Y|X}$ is a BSC(p_A), where $p_A \in [0, 0.5]$. Using Mrs. Gerber's lemma [23], the key-leakage-storage region of the GS model for this case is

$$\begin{aligned} \mathcal{R}_{\text{gs}} = \bigcup_{q \in [0,0.5]} \Big\{ (R_s, R_l, R_w) \colon \quad & 0 \le R_s \le 1 - H_b(q * p_A), \\ & R_l \ge H_b(q * p_A) - H_b(q), \\ & R_w \ge H_b(q * p_A) - H_b(q) \Big\}. \end{aligned} \tag{20}$$

The storage rate of the WZ code construction above is

$$R_w = \frac{\log |\mathcal{W}|}{n} = \frac{m_2}{n} = H_b(q * p_A) - H_b(q) + \delta \tag{21}$$

which follows from (12) and (13). Using (16), the secret-key and privacy-leakage rates are

$$R_s = \frac{H(S)}{n} = \frac{n-m_1}{n} - \frac{1}{n}H(X_q^n\mathbf{H}^T) \tag{22}$$

$$R_l = \frac{H(W)}{n} = \frac{1}{n}H(X_q^n\mathbf{H}^T). \tag{23}$$

Since the parity-check matrix $\mathbf{H}_2$ has full rank, the public side information in (15) is uniformly distributed because X_q^n is uniformly distributed (see also [24, Appendix A]) due to the uniformity of X^n and linearity of $\mathcal{C}_1$. With the assumptions above, and using $X_q^n\mathbf{H}_1^T = 0$, we obtain

$$R_s = \frac{n-m_1}{n} - \frac{m_2}{n} = 1 - H_b(q*p_A) - 2\delta \tag{24}$$

$$R_l = \frac{m_2}{n} = H_b(q*p_A) - H_b(q) + \delta \tag{25}$$

$$R_w = \frac{m_2}{n} = H_b(q*p_A) - H_b(q) + \delta. \tag{26}$$

One can choose a δ in (12) and (13) such that $\delta \to 0$ when $n \to \infty$. Therefore, the boundary points of the key-leakage-storage region $\mathcal{R}_{\text{gs}}$ are the union of (R_s, R_l, R_w) rate tuples over all $q \in [0, 0.5]$.

B. Optimality of CS Model Rates

The key-leakage-storage region of the CS model for a uniform binary source measured through a BSC(p_A) is similarly

$$\begin{aligned}\mathcal{R}_{\text{cs}} = \bigcup_{q\in[0,0.5]} \Big\{ &(R_s, R_l, R_w): \quad 0 \le R_s \le 1 - H_b(q*p_A),\\ &R_l \ge H_b(q*p_A) - H_b(q),\\ &R_w \ge 1 - H_b(q)\Big\}.\end{aligned} \tag{27}$$

The storage rate for this case is the sum of the storage and secret-key rates of the GS model, i.e., we have

$$R_w = \frac{n-m_1}{n} = 1 - H_b(q) - \delta \tag{28}$$

which follows from (24) and (26). The secret-key and privacy-leakage rates are the same as in the GS model, and we have

$$R_s = \frac{n-m_1-m_2}{n} = 1 - H_b(q*p_A) - 2\delta \tag{29}$$

$$R_l = \frac{m_2}{n} = H_b(q*p_A) - H_b(q) + \delta. \tag{30}$$

The union of these rate tuples over all $q \in [0, 0.5]$ includes all boundary points of $\mathcal{R}_{\text{cs}}$.

IV. Comparisons with Existing Methods

There are several existing methods proposed for the CS and GS models, but some methods leak information about the secret key. We consider three methods that satisfy the secrecy leakage constraint in (2): the fuzzy-commitment scheme [11] for the CS model, the code-offset fuzzy extractors [12] for the GS model, and the polar code construction given in [5] for the GS model.

The fuzzy commitment scheme has enrollment and reconstruction steps. During enrollment, an encoder takes a uniformly distributed secret key S' as input to generate a codeword C^n. The codeword and the binary source output X^n are summed modulo-2, and the sum is stored as helper data W'. During reconstruction, W' and another binary sequence Y^n, correlated with X^n through a BSC(p_A), are summed modulo-2 and this sum is used by a decoder to estimate S'. Similar steps are applied in the code-offset fuzzy extractors, except that the secret key is a hashed version of X^n. The fuzzy commitment scheme achieves the Pareto-optimal point in the key-leakage region with the maximum secret-key rate $R_s = I(X;Y)$; the privacy-leakage rate is $R_l = H(X|Y)$ [25]. Similarly, the code-offset fuzzy extractors achieve the same boundary point in the key-leakage region. This is, however, the only Pareto-optimal point that these methods achieve.

We can improve both methods by adding a VQ step: instead of X^n we use its quantized version X_q^n during enrollment. This asymptotically corresponds to summing the original helper data and an independent random variable $J^n \sim \text{Bern}^n(q)$ such that $W'' = X^n \oplus C^n \oplus J^n$ is the new helper data. The modified fuzzy commitment scheme and code-offset fuzzy extractors can achieve all points of the key-leakage region. However, the helper data length n is equal to the length of X^n for both methods. The resulting storage rate of 1 bits/symbol is not optimal.

The polar code construction in [5] without an additional private key requires less storage than the fuzzy commitment scheme and code-offset fuzzy extractors. However, this approach improves only the storage rate and cannot achieve all points of the key-leakage-storage region. One can improve this construction by adding a VQ before the polar encoder during enrollment. However, the effects of quantization on the finite-length code design are not easy to analyze due to the lack of a nested structure in the code construction.

The existing methods cannot, therefore, achieve all points of the key-leakage-storage region for a BSC, unlike the construction in Section III. The construction uses a code $\mathcal{C}_1$ as a good source code to quantize a noiseless identifier output and a subcode $\mathcal{C}$ of $\mathcal{C}_1$ to correct the errors in the noisy identifier outputs. Using these nested codes, the same secret key is obtained during enrollment and reconstruction with high probability, and we can asymptotically achieve all boundary points by taking a union of all achieved rate tuples in (24)-(26) or (28)-(30) over all $q \in [0, 0.5]$.

In previous works such as [26], only the secret-key rates of the proposed codes are compared because the sum of the secret-key and privacy-leakage rates is one. This constraint means that increasing the key vs. leakage rate ratio is equivalent to increasing the key rate. However, the privacy-leakage rate should be infinite to achieve the maximum secret-key rate [27], and our construction is more flexible than the existing methods in terms of achievable rate tuples. We therefore use the key vs. leakage rate ratio as a metric to control the privacy leakage in our finite-length designs below.

V. Proposed Polar Codes for Key Generation

Consider the WZ code construction from [13], which is based on nested polar codes and does not require privacy

amplification, unlike the code construction in Section III. We give an example for a uniform source output and a BSC for the GS model by using polar codes. Polar codes have low encoding and decoding complexity, and it is straightforward to create a nested structure with them.

When designing polar codes, we want to increase the key vs. leakage rate ratio

$$\frac{R_s}{R_l} = \frac{n-m_1}{m_2} - 1 \tag{31}$$

which suggests using small m_1 and m_2. However, large m_1 and m_2 are needed to satisfy the block-error probability constraint, which results in a tradeoff. Note also that for the WZ construction applied to the GS model, the privacy-leakage and storage rates are equal. We thus use the secret-key and privacy-leakage rates for analysis.

A. Polar Codes

Polar codes rely on the *channel polarization* phenomenon, where the physical channel is converted into virtual channels by a polar transform. This transform converts an input sequence U^n with frozen and unfrozen bits to a codeword of the same length n. A polar decoder processes a noisy observation of the codeword together with the frozen bits to estimate U^n [28].

Let $\mathcal{C}(n, \mathcal{F}, G^{|\mathcal{F}|})$ denote a polar code of length n, where $\mathcal{F}$ is the set of indices of the frozen bits, or channels, and $G^{|\mathcal{F}|}$ is the sequence of values of the corresponding frozen bits. In the following, we use the nested polar code construction proposed in [13]. Similar to Section III, this construction achieves the boundary points of the key-leakage-storage region of the GS model for large n. Using an additional one-time pad step with an embedded secret key, one can show optimality also for the CS model.

B. Polar Code Construction for the GS Model

We use two polar codes $\mathcal{C}_1(n, \mathcal{F}_1, V)$ and $\mathcal{C}(n, \mathcal{F}, \overline{V})$ with $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_w$ and $\overline{V} = [V, W]$, where V has length m_1 and W has length m_2. The indices in $\mathcal{F}_1$ represent frozen channels for both codes and $\mathcal{C}$ has some additional frozen channels denoted by $\mathcal{F}_w$, i.e., the codes are nested.

The code $\mathcal{C}_1$ serves as a VQ with a desired distortion q, and the code $\mathcal{C}$ serves as the error correcting code for a BSC($q * p_A$). The idea is to obtain W during enrollment and store it as public helper data. For reconstruction, W is used by the decoder to estimate the secret key S of length $n - m_1 - m_2$. Fig. 4 shows the block diagram of the proposed construction. In the following, suppose V is the all-zero vector so that no additional storage is necessary.

Enrollment: The uniform binary sequence X^n generated by a PUF during enrollment is treated as the noisy observation of a BSC(q). X^n is quantized by a polar decoder of $\mathcal{C}_1$. We extract from the decoder output U^n the bits at indices $\mathcal{F}_w$ and store them as the helper data W. The bits at the indices $i \in \{1, 2, \ldots, n\} \setminus \mathcal{F}$ are used as the secret key. Note that applying a polar transform to U^n generates X_q^n, which is a distorted version of X^n. The distortion between X^n and X_q^n is modeled as a BSC(q) because the error sequence $E_q^n = X^n \oplus X_q^n$ resembles an i.i.d. sequence $\sim \text{Bern}^n(q)$ when $n \to \infty$, as in Section III.

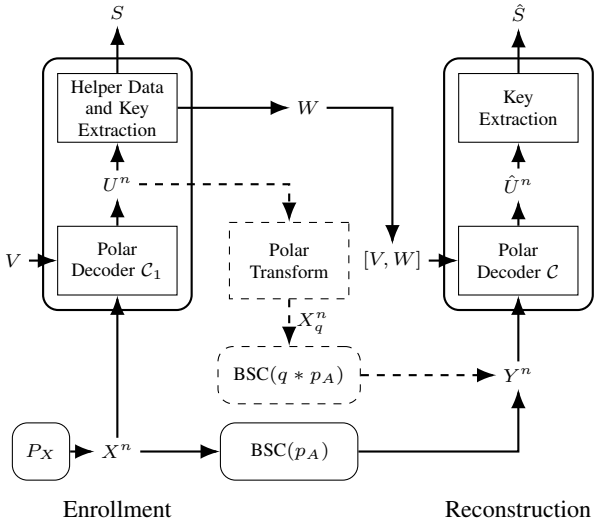

Fig. 4. Proposed polar code based construction for the GS model.

Reconstruction: During reconstruction, the polar decoder of $\mathcal{C}$ observes the binary sequence Y^n (a noisy measurement of X^n through a BSC(p_A) $P_{Y|X}$). The frozen bits $\overline{V} = [V, W]$ at indices $\mathcal{F}$ are input to the polar decoder. The output $\widehat{U}^n$ of the polar decoder is the estimate of U^n and contains the estimate $\widehat{S}$ of the secret key at the unfrozen indices of $\mathcal{C}$, i.e., $i \in \{1, 2, \ldots, n\} \setminus \mathcal{F}$.

Construction of $\mathcal{C}$ and $\mathcal{C}_1$: Since $\mathcal{C} \subseteq \mathcal{C}_1$ are nested codes, they must be constructed jointly. In particular, $\mathcal{F}$ and $\mathcal{F}_1$ should be selected such that the reliability and security constraints are satisfied. For a given secret key size $n - m_1 - m_2$, block length n, crossover probability p_A, and target block-error probability $P_B = \Pr[S \neq \widehat{S}]$, we propose the following procedure.

1) Construct a polar code of rate $(n - m_1 - m_2)/n$ and use it as the code $\mathcal{C}$, i.e., define the set of frozen indices $\mathcal{F}$.
2) Evaluate the error correction performance of $\mathcal{C}$ with a decoder for a BSC over a range of crossover probabilities to obtain the crossover probability p_c, resulting in a target block-error probability of P_B. Using $p_c = E[q] * p_A$, we obtain the target distortion $E[q]$ averaged over all realizations of X^n.
3) Find an $\mathcal{F}_1 \subset \mathcal{F}$ that results in an average distortion of $E[q]$ with a minimum possible amount of helper data. Use $\mathcal{F}_1$ as the frozen set of $\mathcal{C}_1$.

Step 1 is a conventional polar code construction task and step 2 can be applied by Monte-Carlo simulations. For step 3, we start with $\mathcal{F}_1' = \mathcal{F}$ and compute the resulting average distortion $E[q']$ via Monte-Carlo simulations. If $E[q']$ is not less than $E[q]$, we remove elements from $\mathcal{F}_1'$ according to the reliabilities of the polarized bit channels and repeat the procedure until we obtain the desired average distortion $E[q]$.

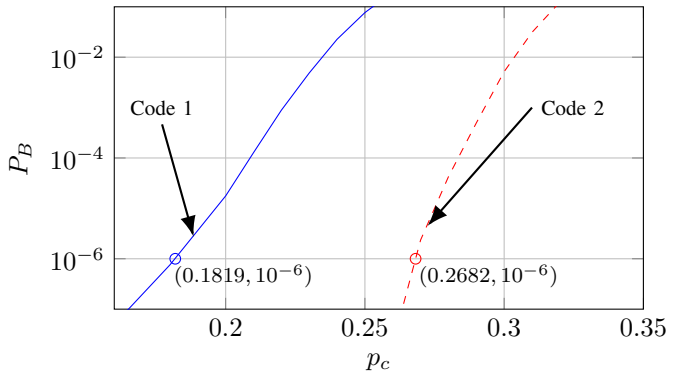

Fig. 5. Block error probability of $\mathcal{C}$ over a BSC(p_c) with an SCL decoder (list size 8) for codes 1 and 2 of length 1024 and 2048, respectively.

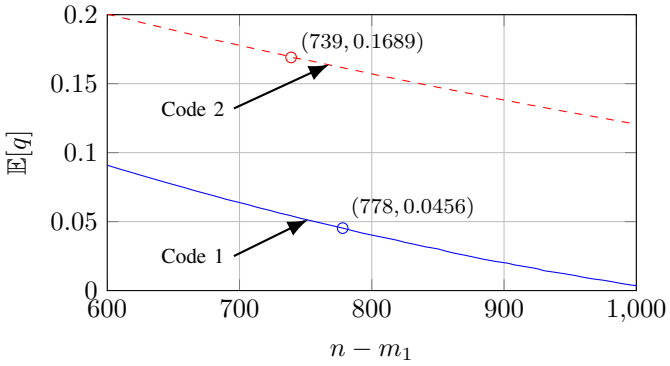

Fig. 6. Average distortion $E[q]$ with respect to $n-m_1$ with an SCL decoder (list size 8) for codes 1 and 2 of length 1024 and 2048, respectively.

C. Proposed Codes for the GS Model

Consider, for instance, the GS model where S is used in the advanced encryption standard (AES) with length 128, i.e., $\log|\mathcal{S}| = n - m_1 - m_2 = 128$ bits. 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} [29]. Consider a BSC $P_{Y|X}$ with crossover probability $p_A = 0.15$, which is a common value for SRAM PUFs. We design nested polar codes for these parameters to illustrate that we can achieve better key-leakage-storage rates than previously proposed codes.

Code 1: Consider $n = 1024$ and recall that $n - m_1 - m_2 = 128$, $P_B = 10^{-6}$, and $p_A = 0.15$. Polar successive cancellation list (SCL) decoders with list size 8 are used as the VQ and channel decoder. We first design the code $\mathcal{C}$ of rate $128/1024$ and evaluate its performance with the SCL decoder for a BSC with a range of crossover probabilities, as shown in Fig. 5. We observe a block-error probability of 10^{-6} at a crossover probability of $p_c = 0.1819$. Since $p_A = 0.15$, this corresponds to an average distortion of $E[q] = 0.0456$, i.e., $E[q] * p_A = 0.1819$.

Fig. 6 shows the average distortion $E[q]$ with respect to $n - m_1 = n - |\mathcal{F}_1|$, obtained by Monte-Carlo simulations. We observe from Fig. 6 that the target average distortion is obtained at $n - m_1 = 778$ bits. Thus, $m_2 = 650$ bits of helper data suffice to obtain a block-error probability of $P_B = 10^{-6}$ to reconstruct a $n - m_1 - m_2 = 128$-bits secret key.

Remark 1. Observe that the parameter p_c is less than $p_A = 0.15$ when we apply the procedure in Section V-B to $n = 512$ with the same P_B. Therefore, it is not possible to construct a code with this procedure for $n \leq 512$ since $q * p_A$ is an increasing function of q for any $q \in [0, 0.5]$. Such a code construction for $n = 512$ might be possible if one improves the decoder.

Code 2: Consider the same parameters as in code 1, except $n = 2048$. We apply the same steps as above and plot the performance of an SCL decoder for a BSC with a range of crossover probabilities in Fig. 5. A crossover probability of $p_c = 0.2682$ is required to obtain a block-error probability of 10^{-6}, which gives an average distortion of $E[q] = 0.1689$. Fig. 6 shows the average distortion with respect to $n - m_1$. We achieve the target average distortion with $n - m_1 = 739$ bits so that helper data of length 611 is required to satisfy $P_B = 10^{-6}$ for a secret key of length 128 bits.

The error probability P_B is here an average over all PUF realizations, i.e., over all PUF devices with the same circuit design. To satisfy this block-error probability for each PUF realization, one could consider using the maximum distortion instead of $E[q]$ as a metric in step 3 above. This would increase the amount of helper data. We can guarantee a block-error probability of at most 10^{-6} for 99.99% of all realizations x^n of X^n by adding 32 bits to the helper data for code 1 and 33 bits for code 2. The numbers of extra helper data bits required are small since the variance of the distortion q over all PUF realizations is small. For comparisons, we use the helper data sizes required to guarantee $P_B = 10^{-6}$ for 99.99% of all PUF realizations.

D. Code Comparisons and Discussions

We achieve the key-leakage-storage rates $(0.1250, 0.6660, 0.6660)$ bits/symbol by code 1 and $(0.0625, 0.3145, 0.3145)$ bits/symbol by code 2. These rates are significantly better than the best rate tuple $(0.1250, 0.8750, 0.8750)$ in [5] in the literature for the same parameters without any private key requirement. Moreover, we increase the key vs. leakage rate ratio from 0.1877 for code 1 to 0.1988 for code 2 by using longer blocklengths, which suggests to further increase the blocklength to obtain better ratios.

Code 2 achieves a privacy-leakage rate that cannot be achieved by the existing methods without applying *time sharing* (see, e.g., [16, Sec. 4.4] for its definition). This is because the privacy-leakage rate 0.3145 bits/symbol achieved by code 2 is significantly less than the minimum privacy-leakage rate $R_l = H_b(p_A) = 0.6098$ bits/symbol achieved by, e.g., the method of [11] without time sharing between the all-zero point and the point with the maximum secret-key rate and minimum possible privacy-leakage rate.

Consider, for instance, again the operating point with the maximum secret-key rate and minimum possible privacy-leakage and storage rates. This point can be asymptotically achieved by the fuzzy commitment scheme. However, the

sphere packing bound (also the best possible meta-converse bound for the source and channel we use [30]) states that the highest key vs. leakage rate ratio that can be achieved at this operating point with $p_A = 0.15$, $n = 1024$, and $P_B = 10^{-6}$ is $R_s/R_l = 0.3750$. Similarly, the ratio is $R_s/R_l = 0.4373$ for $n = 2048$ with the same p_A and P_B. These results indicate that there are still gaps between the maximum finite-length key vs. leakage rate ratios and the ratios 0.1877 and 0.1988 achieved, respectively, by codes 1 and 2 above. The implementation gaps can be reduced by using larger list sizes at the decoder, which is not desired for IoT applications where hardware complexity is limited. For other applications, codes that satisfy $P_B \leq 10^{-9}$ should be designed, for which either laborious decoder simulations or analytical block-error probability bounds seem to be required.

VI. Conclusion

There are codes, encoders, and decoders that asymptotically achieve all points of the rate regions of the WZ and GS problems simultaneously. Extending the functional equivalence, we applied two asymptotically optimal WZ code constructions to the GS and CS models for binary sources measured through a BSC. We designed finite-length nested polar codes that achieve better rate tuples than existing methods, and one of our codes achieves a rate tuple that cannot be achieved by existing methods without time sharing. Gaps to the maximum key vs. leakage rate ratios were illustrated. In future work, we will design nested polar codes with a block-error probability $P_B \leq 10^{-9}$ by approximating the weight distributions of the codes.

Acknowledgment

The authors thank Amin Gohari for his insightful comments and Peihong Yuan for his help with polar codes.

References

[1] O. Günlü, O. İşcan, and G. Kramer, "Reliable secret key generation from physical unclonable functions under varying environmental conditions," in *IEEE Int. Workshop Inf. Forensics Security*, Rome, Italy, Nov. 2015, pp. 1–6.

[2] T. Ignatenko and F. M. J. Willems, "Biometric systems: Privacy and secrecy aspects," *IEEE Trans. Inf. Forensics Security*, vol. 4, no. 4, pp. 956–973, Dec. 2009.

[3] B. Gassend, "Physical random functions," Master's thesis, M.I.T., Cambridge, MA, Jan. 2003.

[4] R. Pappu, "Physical one-way functions," Ph.D. dissertation, M.I.T., Cambridge, MA, Oct. 2001.

[5] B. Chen, T. Ignatenko, F. M. Willems, R. Maes, E. van der Sluis, and G. Selimis, "A robust SRAM-PUF key generation scheme based on polar codes," July 2017, [Online]. Available: arxiv.org/abs/1701.07320.

[6] M. A. Ferrag, L. Maglaras, A. Argyriou, D. Kosmanos, and H. Janicke, "Security for 4G and 5G cellular networks: A survey of existing authentication and privacy-preserving schemes," Aug. 2017, [Online]. Available: arxiv.org/abs/1708.04027.

[7] L. Lai, S.-W. Ho, and H. V. Poor, "Privacy-security trade-offs in biometric security systems - Part I: Single use case," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 1, pp. 122–139, Mar. 2011.

[8] M. Koide and H. Yamamoto, "Coding theorems for biometric systems," in *IEEE Int. Symp. Inf. Theory*, Austin, TX, June 2010, pp. 2647–2651.

[9] O. Günlü and G. Kramer, "Privacy, secrecy, and storage with noisy identifiers," Jan. 2016, [Online]. Available: arxiv.org/abs/1601.06756.

[10] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. 19, no. 4, pp. 471–480, July 1973.

[11] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in *ACM Conf. Comp. and Commun. Security*, New York, NY, Nov. 1999, pp. 28–36.

[12] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM J. Comput.*, vol. 38, no. 1, pp. 97–139, Jan. 2008.

[13] S. B. Korada and R. Urbanke, "Polar codes for Slepian-Wolf, Wyner-Ziv, and Gelfand-Pinsker," in *IEEE Inf. Theory Workshop*, Cairo, Egypt, Jan. 2010, pp. 1–5.

[14] A. Orlitsky and J. R. Roche, "Coding for computing," *IEEE Trans. Inf. Theory*, vol. 47, no. 3, pp. 903–917, Mar. 2001.

[15] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inf. Theory*, vol. 22, no. 1, pp. 1–10, Jan. 1976.

[16] A. E. Gamal and Y.-H. Kim, *Network Information Theory*. Cambridge, U.K.: Cambridge Uni. Press, 2011.

[17] S. S. Pradhan, J. Chou, and K. Ramchandran, "Duality between source coding and channel coding and its extension to the side information case," *IEEE Trans. Inf. Theory*, vol. 49, no. 5, pp. 1181–1203, May 2003.

[18] S. I. Gel'fand and M. S. Pinsker, "Coding for channels with random parameters," *Probl. Contr. Inf. Theory*, vol. 9, no. 1, pp. 19–31, Jan. 1980.

[19] S. Shamai, S. Verdú, and R. Zamir, "Systematic lossy source/channel coding," *IEEE Trans. Inf. Theory*, vol. 44, no. 2, pp. 564–579, Mar. 1998.

[20] J. Wayman, A. Jain, D. Maltoni, and D. M. (Eds), *Biometric Systems: Technology, Design and Performance Evaluation*. London, U.K.: Springer-Verlag, 2005.

[21] M. Bloch and J. Barros, *Physical-layer Security*. Cambridge, U.K.: Cambridge Uni. Press, 2011.

[22] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*. Mineola: NY: Dover Publications, 2013.

[23] A. D. Wyner and J. Ziv, "A theorem on the entropy of certain binary sequences and applications: Part I," *IEEE Trans. Inf. Theory*, vol. 19, no. 6, pp. 769–772, Nov. 1973.

[24] Y. Wang, S. Rane, S. C. Draper, and P. Ishwar, "A theoretical analysis of authentication, privacy, and reusability across secure biometric systems," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 6, pp. 1825–1840, July 2012.

[25] T. Ignatenko and F. M. J. Willems, "Information leakage in fuzzy commitment schemes," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 2, pp. 337–348, Mar. 2010.

[26] R. Maes, A. V. Herrewege, and I. Verbauwhede, "PUFKY: A fully functional PUF-based cryptographic key generator," in *Cryptographic Hardware Embedded Sys.* Berlin Heidelberg, Germany: Springer-Verlag, Sep. 2012, pp. 302–319.

[27] T. Ignatenko and F. M. J. Willems, "Privacy-leakage codes for biometric authentication systems," in *IEEE Int. Conf. Acoustics, Speech Signal Process.*, May 2014, pp. 1601–1605.

[28] E. Arikan, "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, July 2009.

[29] C. Bösch, J. Guajardo, A.-R. Sadeghi, J. Shokrollahi, and P. Tuyls, "Efficient helper data key extractor on FPGAs," *Cryptographic Hardware Embedded Syst.*, pp. 181–197, Aug. 2008.

[30] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.