Privacy, Secrecy, and Storage with Nested Randomized Polar Subcode Constructions

Onur Günlü, *Member, IEEE,* Peter Trifonov, *Member, IEEE,*
Muah Kim, *Student Member, IEEE,* Rafael F. Schaefer, *Senior Member, IEEE,*
and Vladimir Sidorenko, *Member, IEEE*

Abstract

We consider a set of security and privacy problems under reliability and storage constraints that can be tackled by using codes and particularly focus on the secret-key agreement problem. Polar subcodes

O. Günlü, M. Kim, and R. F. Schaefer were supported by the German Federal Ministry of Education and Research (BMBF) within the national initiative for *Post Shannon Communication (NewCom)* under the Grant 16KIS1004. P. Trifonov was supported by the Government of Russian Federation under the Grant 08-08. V. Sidorenko is on leave from the Institute for Information Transmission Problems, Russian Academy of Science. His work was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant Agreement No: 801434) and by the Institute of Communications Engineering at TU Munich. Parts of this paper were presented at the 2020 IEEE International Symposium on Information Theory and Applications in [1].

O. Günlü and M. Kim are with the Information Theory and Applications Chair, Technische Universität Berlin, 10623 Berlin, Germany (E-mail: {guenlue, muah.kim}@tu-berlin.de).

P. Trifonov is with the Faculty of Secure Information Technologies, ITMO University, 197101 Saint Petersburg, Russia (E-mail: pvtrifonov@itmo.ru).

R. F. Schaefer is with the Chair of Communications Engineering and Security, University of Siegen, 57076 Siegen, Germany (E-mail: rafael.schaefer@uni-siegen.de).

V. Sidorenko is with the Institute of Communications Engineering, Technical University of Munich, 80333 Munich, Germany (E-mail: vladimir.sidorenko@tum.de).

(PSCs) are polar codes (PCs) with dynamically-frozen symbols and have a larger code minimum distance than PCs with only statically-frozen symbols. A randomized nested PSC construction, where the low-rate code is a PSC and the high-rate code is a PC, is proposed for successive cancellation list (SCL) and sequential decoders. This code construction aims to perform lossy compression with side information, i.e., Wyner-Ziv (WZ) coding. Nested PSCs are used in the key agreement problem with physical identifiers and two terminals since WZ-coding constructions significantly improve on Slepian-Wolf coding constructions such as fuzzy extractors. Significant gains in terms of the secret-key vs. storage rate ratio as compared to nested PCs with the same list sizes are illustrated to show that nested PSCs significantly improve on all existing code constructions. The performance of the nested PSCs is shown to improve with larger list sizes, unlike the nested PCs considered. A design procedure to efficiently construct nested PSCs and possible improvements to the nested PSC designs are also provided.

Index Terms

polar subcodes, sequential decoding, coding for privacy, physical unclonable functions (PUFs), list decoding.

I. Introduction

A common secrecy problem considers the wiretap channel (WTC) [2]. The WTC encoder aims to hide a transmitted message from an eavesdropper with a channel output correlated with the observation of a legitimate receiver. There are various code constructions for the WTC that achieve the secrecy capacity, e.g., in [3]–[7]. Some of these constructions use nested polar codes (PCs) because PCs have a low encoding/decoding complexity, asymptotic optimality for various problems [8], and good finite length performance if a successive cancellation list (SCL) decoder in combination with an outer cyclic redundancy check (CRC) code are used [9]. Similarly, nested PCs achieve the strong coordination capacity boundaries [10].

A closely related secrecy problem to the WTC problem is the key agreement problem with two terminals that observe correlated random variables and have access to a public, authenticated,

and one-way communication link, whereas an eavesdropper observes only the public messages called *helper data* [11], [12]. There are two common models for key agreement: the *generated-secret (GS)* model, where an encoder extracts a secret key from the sequence observed, and the *chosen-secret (CS)* model, where a pre-determined secret key is given as input to the encoder, respectively. The main constraint for this problem is that the code construction should not leak information about the secret key (negligible *secrecy leakage*). Furthermore, a *privacy leakage* constraint is introduced in [13] to leak as little information about the identifier as possible. Similarly, *storage* in the public communication link can be expensive and limited, e.g., for internet-of-things (IoT) device applications [14], [15]. The regions of achievable secret-key vs. privacy-leakage (or key-leakage) rates for the GS and CS models are given in [13], while the key-leakage-storage regions with multiple encoder measurements are treated in [16].

An important application of these key agreement models is the key agreement with physical identifiers such as digital circuits that have outputs unique to the device that embodies them. Examples of these physical identifiers are physical unclonable functions (PUFs) [17]–[19]. The start-up behavior of static random access memories (SRAM) and the speckle pattern observed from coherent waves propagating through a disordered medium can serve as PUFs that have reliable outputs and high entropy [20], [21].

Optimal nested random linear code constructions for the lossy source coding with side information problem, i.e., Wyner-Ziv (WZ) problem [22], are shown in [15] to be optimal also for the key agreement with PUFs. Furthermore, WZ-coding constructions are proved in [15], [23] to significantly improve on Slepian-Wolf (SW) coding [24] constructions used for key agreement. Thus, nested PCs are designed in [15] for practical SRAM PUF parameters to illustrate that nested PCs achieve rate tuples that cannot be achieved by using previous code constructions. The finite length performance of the nested PCs designed in [15] without an outer CRC code is not necessarily good due to small minimum distance of PCs. Therefore, we propose to increase the code minimum distance by using PCs with dynamically-frozen symbols (DFSs), i.e., *polar*

subcodes (PSCs) [25].

PSCs assign a set of dynamic freezing constraints such that linear combinations of other symbols are assigned to DFSs, rather than zeros assigned to statically-frozen symbols (SFSs) for PCs. The set of linear equations can be chosen such that the resulting codewords are a subcode of a parent code with large code minimum distance such as extended Bose–Chaudhuri–Hocquenghem (eBCH) codes [25]. Polar subcodes of eBCH codes require a large list size to approach the maximum likelihood (ML) decoding performance. This is not desirable for the key agreement problem with SRAM PUFs because, e.g., the list size of the nested PCs used in [15] is 8, which suffices to approach the ML decoding performance, and a larger list size might result in a high hardware cost. Therefore, we use a randomized PSC construction from [26] with two types of DFSs. The first type of symbols are called *type-A DFSs*, eliminating the low-weight codewords, and the second type are called *type-B DFSs*, hindering the correct path to be killed by the decoding algorithm. The randomized construction has good performance for a list size of 8 and its performance improves with larger list sizes, as illustrated below.

A. Identifier Output Models

We consider identifier outputs, such as biometric or physical identifier outputs, that are independent and identically distributed (i.i.d.) according to a probability distribution over a discrete alphabet. There are various ways to extract almost i.i.d. symbols from identifiers, one of which is to apply transform-coding algorithms to decorrelate the raw identifier outputs, as in [27]–[30]. Therefore, our identifier output models are realistic.

B. Summary of Contributions

We design codes for key agreement with PUFs by constructing nested PSCs in a randomized manner. Nested codes have a broad use, e.g., in WTC and strong coordination problems, so the proposed nested PSC constructions can be useful also for these problems. A summary of the

main contributions is as follows, where the last three contributions are novel ones that are not mentioned in the conference version of this work in [1].

- We propose a method to obtain nested PSCs used as a WZ-coding construction, which is a binning method that can be useful for various information-theoretic problems. Furthermore, we develop a step-by-step design procedure for the proposed nested PSC construction adapted to the problem of key agreement with physical identifiers. We consider binary symmetric sources (BSSs) and binary symmetric channels (BSCs). Ring oscillator (RO) PUFs combined with transform coding [31] and SRAM PUFs [32], [33] are modeled by these sources and channels.
- We design and simulate nested PSCs for practical source and channel parameters for SRAM PUFs, as in [15]. We illustrate that all designed nested PSCs with sequential decoders for a list size L of 8 achieve a significantly larger key vs. storage rate ratio than all previously-proposed code constructions including nested PCs from [15] that approach its maximum likelihood (ML) performance with an SCL decoder for $L=8$. Nested PSC performance is illustrated to further improve with larger but reasonable list sizes such as 32 and 64.
- To take advantage of the significantly better performance of nested PSCs as compared to nested PCs, we design and simulate nested PSCs to find the smallest list sizes for which at least the same performance as being achieved by state-of-the-art code constructions can be achieved. For a secret key size of 128 bits, block-error probability 10^{-6}, and blocklength 1024 bits, we show that it suffices to have a list of size of $L = 4$ for nested PSCs rather than $L = 8$ that is required by state-of-the-art nested PCs. Similarly, for the same design parameters except that the blocklength is 2048 bits, nested PSCs are shown to require only $L = 6$ rather than $L = 8$. These linear reductions in the list sizes suggest a linear hardware cost gain for the (high-rate) PC as the effect of the list size on the complexity is known to be approximately linear for PCs. The hardware cost gain due to smaller list sizes is known

to be sublinear for (low-rate) PSCs, but the hardware cost of PSCs are considered to be smaller than of PCs.

- We provide the average number of summation and comparison operations done in the sequential decoders of the low-rate PSC and high-rate PCs that are designed for SRAM PUFs by using our design procedure proposed for nested PSCs. The results illustrate that the high-rate PCs dominate the overall complexity and their complexity depends significantly on the list size, whereas the changes in the complexity of the low-rate PSCs with respect to changing list sizes are negligible.
- The most promising and effective parameters, used in the proposed nested PSC design procedure, that might be further optimized to improve the privacy, secrecy, reliability, storage or hardware cost performance, are provided. These parameters and the methods suggested to further optimize them are based on both numerous simulations conducted and fundamental properties of PSCs.

C. Organization

This paper is organized as follows. In Section II, we describe the GS and CS models, and evaluate the key-leakage-storage region for BSSs and BSCs. We propose a randomized nested PSC construction and a design procedure adapted to key agreement with PUFs in Section III. Significant key vs. storage rate ratio gains obtained from nested PSCs designed for practical SRAM PUF parameters as compared to previously-proposed codes are illustrated in Section IV. In Section V, we discuss how to improve the performance of the designed nested PSCs by tuning the design parameters up. Section VI concludes the paper.

D. Notation

Upper case letters represent random variables and lower case letters their realizations. A superscript denotes a string of variables, e.g., $X^n = X_1, X_2, \ldots, X_i, \ldots, X_n$, and a subscript i

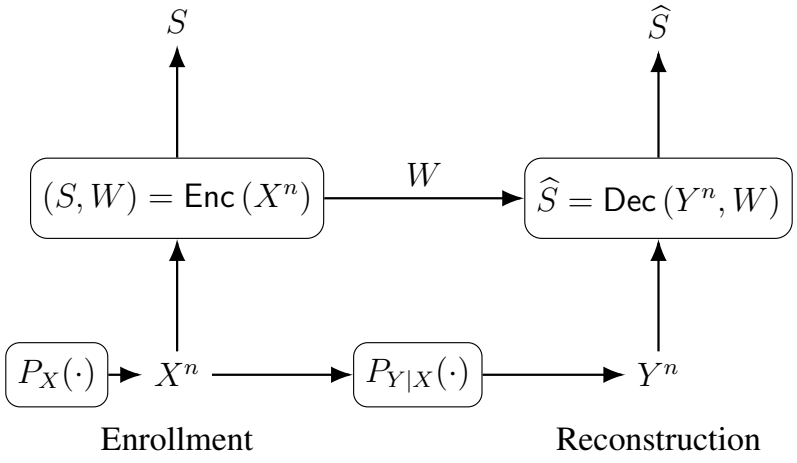

Fig. 1. The GS model, where a secret is generated by the encoder.

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, set sizes are written as $|\mathcal{X}|$. $H_b(x) = -x \log x - (1-x) \log(1-x)$ is the binary entropy function, where logarithms are to the base 2, and $H_b^{-1}(\cdot)$ denotes its inverse with range $[0, 0.5]$. The star operation is defined as $q * p = (1-2p)q + p$ with its inverse operation $q = \dfrac{(q * p) - p}{1 - 2p}$. A BSC with crossover probability p is denoted by BSC(p). $X \sim \text{Bern}(\alpha)$ is a binary random variable with $\Pr[X = 1] = \alpha$. For integers $n \geq 1$ and $j_1, j_2 \in \{0, 1, \ldots, n-1\}$, $\{0, 1, \ldots, n-1\} \setminus \{j_1, j_2\}$ denotes the set $\{0, 1, \ldots, j_1 - 1, j_1 + 1, \ldots, j_2 - 1, j_2 + 1, \ldots, n-1\}$. Given matrices $\mathbb{V}$ and $\mathbb{V}'$, $\mathbb{V}^{\otimes m} = \mathbb{V} \otimes \mathbb{V}^{\otimes(m-1)}$ represents the m-th Kronecker power of matrix $\mathbb{V}$ for all $m \geq 1$, where $\mathbb{V} \otimes \mathbb{V}'$ represents the Kronecker product of the matrices $\mathbb{V}$ and $\mathbb{V}'$ and we define that $\mathbb{V}^{\otimes 0}$ is equal to the value 1.

II. Problem Formulation

An identifier output is used to generate a secret key in the GS model, depicted in Fig. 1. The source $\mathcal{X}$, noisy measurement $\mathcal{Y}$, secret key $\mathcal{S}$, and storage $\mathcal{W}$ alphabets are finite sets. During enrollment, the encoder $\mathsf{Enc}(\cdot)$ observes an i.i.d. identifier output X^n and computes a secret key $S \in \mathcal{S}$ and public helper data $W \in \mathcal{W}$ as $(S, W) = \mathsf{Enc}(X^n)$. During reconstruction, the decoder

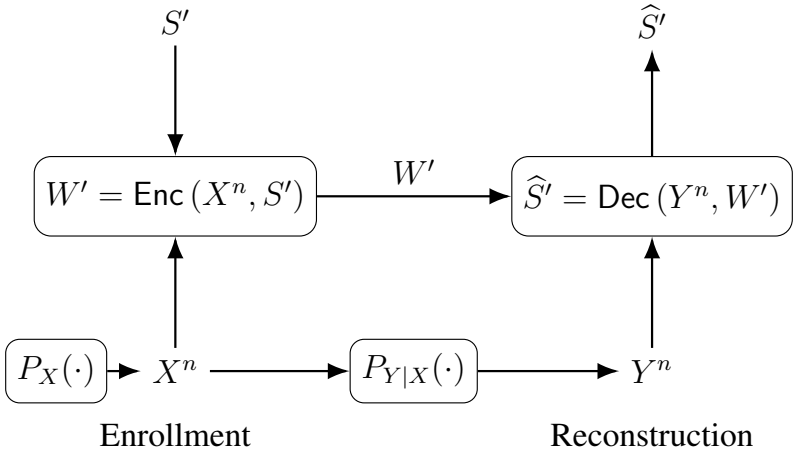

Fig. 2. The CS model, where a chosen secret is embedded into the encoder.

$\mathsf{Dec}(\cdot)$ observes a noisy source measurement Y^n of the source output X^n through a memoryless measurement channel $P_{Y|X}$ in addition to the helper data W. The decoder estimates the secret key as $\widehat{S} = \mathsf{Dec}(Y^n, W)$.

Fig. 2 shows the CS model, where a pre-determined secret key $S' \in \mathcal{S}$ that is mutually independent of (X^n, Y^n) is embedded into the helper data as $W' = \mathsf{Enc}(X^n, S')$. The decoder for the CS model estimates the secret key as $\widehat{S}' = \mathsf{Dec}(Y^n, W')$. The CS model is illustrating the cases where the encoder, such as a trusted entity or a manufacturer, chooses the secret key according to other practical constraints. The CS model can be equivalently represented as the GS model whose generated secret key S is summed in modulo-$|\mathcal{S}|$ with the pre-determined secret key S', which is a Vernam cipher [34]. This representation of the CS model makes its achievability proof straightforward given the achievability proof of the GS model [11], [13], [21]. Since the analyses for the CS model follows straightforwardly from the analyses for the GS model, it suffices to consider the GS model to illustrate the performance gains from nested PSCs.

Definition 1. *A key-leakage-storage tuple* (R_s, R_ℓ, R_w) *is* achievable *for the GS model if, given*

any $\epsilon > 0$*, there is some* $n \geq 1$*, an encoder, and a decoder such that* $R_s = \dfrac{\log |\mathcal{S}|}{n}$ *and*

$$P_B \triangleq \Pr[\widehat{S} \neq S] \leq \epsilon \qquad (\textit{reliability}) \tag{1}$$

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

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

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

$$\frac{1}{n} I(X^n;W) \leq R_\ell + \epsilon \qquad (\textit{privacy}). \tag{5}$$

The key-leakage-storage *region* $\mathcal{R}_{gs}$ *for the GS model is the closure of the set of achievable tuples.* ◊

We remark that the secrecy criterion given in (2) provides a weak secrecy guarantee due to the normalization by the blocklength n. Given a code construction that achieves weak secrecy and using information reconciliation and privacy amplification steps in combination with multiple identifier output blocks, as described in [35], one can achieve strong secrecy, for which the unnormalized secrecy leakage $I(S;W)$ is negligibly small. It is discussed in [15] that multiple identifier blocks can be obtained, respectively, by using multiple PUFs in each device for key agreement with physical identifiers and by using multiple biometrics for key agreement with biometric identifiers. Thus, below we consider only weak secrecy for simplicity.

Theorem 1 ([13])**.** *The key-leakage-storage region for the GS model is*

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

where U is an auxiliary random variable, which should be optimized for given P_X and $P_{Y|X}$, such that $U-X-Y$ forms a Markov chain. $\mathcal{R}_{gs}$ is a convex set. It suffices to have $|\mathcal{U}| \leq |\mathcal{X}|+1$.

Suppose the low-complexity transform-coding algorithm in [36] is applied to a PUF circuit with continuous-valued outputs to obtain X^n that is almost i.i.d. according to a uniform Bernoulli random variable, i.e., $X^n \sim \text{Bern}^n(\frac{1}{2})$, and the channel $P_{Y|X}$ is a BSC(p_A) for $p_A \in [0, 0.5]$. We then obtain the following key-leakage-storage region for this identifier output model, which is a result of optimizing the auxiliary random variable U defined in Theorem 1 for these P_X and $P_{Y|X}$ by using Mrs. Gerber's lemma [37].

Corollary 1 ([13])**.** *The key-leakage-storage region $\mathcal{R}_{gs,binary}$ of the GS model for $X^n \sim Bern^n(\frac{1}{2})$ and $P_{Y|X} \sim BSC(p_A)$ is the union over all $q \in [0, 0.5]$ of the bounds*

$$0 \leq R_s \leq 1 - H_b(q * p_A) \tag{7}$$

$$R_\ell \geq H_b(q * p_A) - H_b(q) \tag{8}$$

$$R_w \geq H_b(q * p_A) - H_b(q). \tag{9}$$

The rate tuples on the boundary of the region $\mathcal{R}_{\text{gs,binary}}$ are uniquely defined by the key vs. storage rate ratio $\frac{R_\text{s}}{R_\text{w}}$. We therefore use this ratio as the metric to compare our nested PSCs with previously-proposed nested PCs and other channel codes. A larger key vs. storage rate ratio suggests that the code construction is closer to an achievable point that is on the boundary of $\mathcal{R}_{\text{gs,binary}}$, which is an optimal tuple.

III. Design of Nested PSCs

Polar codes convert a channel into polarized virtual bit channels by a polar transform. This transform converts an input sequence U^n with frozen and unfrozen bits to a length-n codeword. A polar decoder processes a noisy codeword together with the frozen bits to estimate U^n. Let $\mathcal{C}(n, \mathcal{F}, G^{|\mathcal{F}|})$ denote a PC or a PSC of length n, where $\mathcal{F}$ is the set of indices of the frozen

bits and $G^{|\mathcal{F}|}$ is the sequence of frozen bits. In the following, we extend the nested binary PC construction proposed in [38] for the WZ problem.

A. Polar Subcodes and Randomized Construction

PSCs are a generalization of PCs and they allow some frozen symbols to be equal to linear combinations of other symbols [25]. Such symbols are referred as dynamically-frozen symbols (DFSs). An $(n = 2^m, k)$ PSC is defined by an $(n-k) \times n$ constraint matrix $\mathbb{V}$ such that the last non-zero elements of its rows are located in distinct columns $j_i \in \{0, \ldots, n-1\}$ for $0 \leq i < n-k$. The codewords of the polar subcode are obtained as

$$c^n = u^n \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{\otimes m} \tag{10}$$

where the values $G^{|\mathcal{F}|}$ of frozen symbols of u^n are calculated as

$$u_{j_i} = \sum_{s=0}^{j_i - 1} \mathbb{V}_{is} u_s. \tag{11}$$

Decoding of PSCs can be implemented by a successive cancellation (SC) algorithm, as well as its list and sequential decoding generalizations [9], [39]. A simple way to obtain binary PSCs with good performance under list or sequential decoding with small list sizes is to employ a randomized construction introduced in [26]. The construction involves three types of frozen symbols:

- The indices of statically-frozen symbols (SFSs), which are a special case of DFSs, are selected as integers j_i, for $0 \leq i < n-k-t_{\mathrm{A}}-t_{\mathrm{B}}$, of the least reliable virtual subchannels of the polar transform, so the i-th row of $\mathbb{V}$ has 1 in position j_i and 0, otherwise. This corresponds to constraints $u_{j_i} = 0$.
- The indices of type-B DFSs are selected as the integers j_i, for $n-k-t_{\mathrm{A}}-t_{\mathrm{B}} \leq i < n-k-t_{\mathrm{A}}$, of the least reliable virtual subchannels that are not selected as SFSs. The i-th row of $\mathbb{V}$ has 1 in position j_i and binary uniformly-randomly chosen values in positions $s < j_i$. Type-B

DFSs enforce the scores of incorrect paths in the Tal-Vardy decoding algorithm to decrease fast, reducing the probability of the correct path being dropped from the list.

- The indices of type-A DFSs j_i, for $n-k-t_{\mathrm{A}} \leq j_i < n-k$, are selected as the largest integers in $\{0, 1, \ldots, n-1\} \setminus \{j_0, \ldots, j_{n-k-t_{\mathrm{A}}-1}\}$ that have the smallest Hamming weight, which is defined as the number of non-zero bits in a sequence's binary representation. The i-th row of $\mathbb{V}$ has a 1 in position j_i and binary uniformly-randomly chosen values in positions $s < j_i$. Type-A DFSs eliminate the low-weight codewords.

The number t_{A} of type-A DFSs and the number t_{B} of type-B DFSs should be chosen in general via extensive simulations. For simplicity, we use the suggested parameters for $L = 32$ in [40], i.e., we choose

$$t_{\mathrm{A}} = \min\{m,\ (n-k)\} \tag{12}$$

$$t_{\mathrm{B}} = \max\Big\{0,\ \min\{(64 - t_{\mathrm{A}}),\ (n-k-t_{\mathrm{A}})\}\Big\}. \tag{13}$$

To obtain the reliabilities of the subchannels of the polar transform, we use the min-sum density evolution algorithm [41] over a BSC(p), where the crossover probability p is a design parameter to be optimized in general by simulations. One parameter used in the sequential decoder is the priority queue size D [39], for which we use $D = 1024$.

B. Randomized Nested PSC Construction

PCs, including PSCs, provide a simple nested code design due to the control on the subsets of codewords by changing the frozen bits. We first summarize the nested code construction method proposed in [15] for PCs and then extend it to PSCs. We also provide a procedure to design nested binary PSCs for key agreement with PUFs.

1) Nested PC Construction: For the GS model with source and channel models given in Corollary 1, consider two PCs $\mathcal{C}_1(n, \mathcal{F}_1, V)$ 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 such that m_1 and m_2 satisfy [15]

$$\frac{m_1}{n} = H_b(q) - \delta \tag{14}$$

$$\frac{m_1 + m_2}{n} = H_b(q * p_A) + \delta \tag{15}$$

for some distortion $q \in [0, 0.5]$ as in (7)-(9) and any $\delta > 0$.

We remark that (14) implies a vector quantization (VQ) code $\mathcal{C}_1$ that can achieve an average per-letter distortion of at most q when $n \to \infty$ since its code rate is greater than the rate-distortion function $I(X; X_q) = 1 - H_b(q)$ at distortion q, where X_q^n represents the quantized version of the sequence X^n. Furthermore, (15) implies an error-correcting code (ECC) $\mathcal{C}$ that can achieve a negligible error probability for a BSC$(q * p_A)$ when $n \to \infty$ since its code rate is smaller than the channel capacity $I(X_q; Y) = 1 - H_b(q * p_A)$.

During enrollment, the encoder treats the uniform binary sequence X^n as a noisy observation measured through a BSC(q). Decoder of the PC $\mathcal{C}_1$ quantizes X^n to a codeword X_q^n of $\mathcal{C}_1$. Applying the inverse polar transform to X_q^n, the encoder calculates U^n and its bits at indices $\mathcal{F}_w$ are stored as the helper data W. Furthermore, the bits at the indices $i \in \{1, 2, \ldots, n\} \setminus \mathcal{F}$ are used as the secret key S that has a length of $n - m_1 - m_2$. During reconstruction, the decoder of the PC $\mathcal{C}$ observes the helper data W and the binary noisy sequence Y^n. The frozen bits $\overline{V} = [V, W]$ at indices $\mathcal{F}$ and Y^n are input to the PC decoder to obtain the codeword $\widehat{X}_q^n$. Applying the inverse polar transform to $\widehat{X}_q^n$, we obtain $\widehat{U}^n$ that contains the estimate $\widehat{S}$ of the secret key at the indices $i \in \{1, 2, \ldots, n\} \setminus \mathcal{F}$. This nested PC construction is depicted in Fig. 3.

2) Nested PSC Construction: We next extend the nested PC construction to a nested PSC construction and provide also exact design parameters. We observe from simulations that the VQ performance of PSCs are entirely similar to the performance of PCs, so we use a PC as the code $\mathcal{C}_1$ and use a PSC as the code $\mathcal{C}$ due to high gains obtained from PSCs in error correction as compared to PCs. Let $\mathbb{V}_S'$ be the constraint matrix for the code $\mathcal{C}_1$, i.e., $\mathbb{V}_S'$ contains unit vectors with 1s in positions $\mathcal{F}_1$. Then, we ensure that the low-rate PSC $\mathcal{C}$ has SFSs in indices

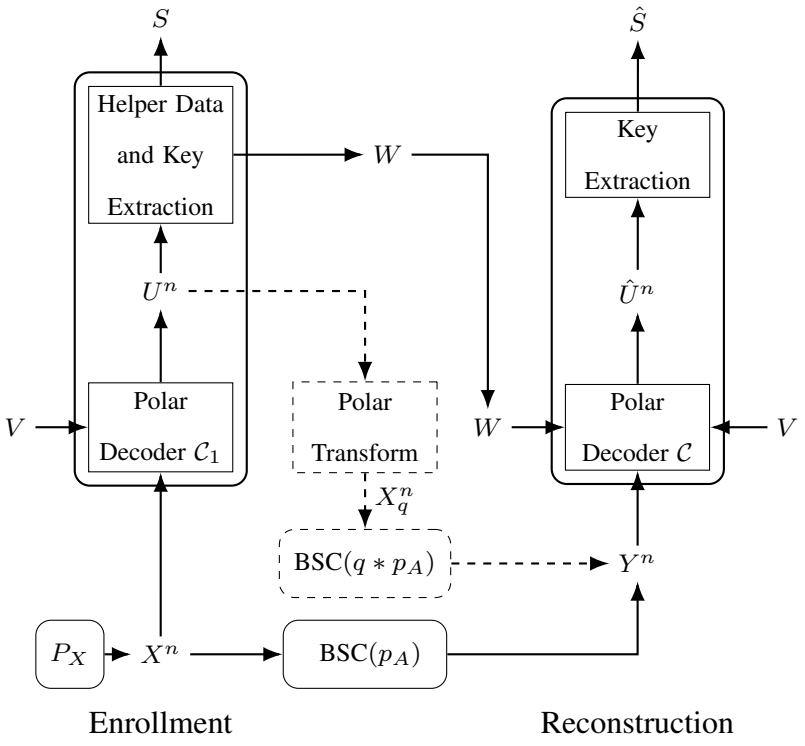

Fig. 3. Nested PC construction for the GS model [15].

$\mathcal{F}_1$. Hence, the constraint matrix $\mathbb{V}$ of $\mathcal{C}$ is given by

$$\mathbb{V} = \begin{bmatrix} \mathbb{V}_S' \\ \mathbb{V}_S'' \\ \mathbb{V}_B \\ \mathbb{V}_A \end{bmatrix} \tag{16}$$

where $\mathbb{V}_A$ and $\mathbb{V}_B$ are submatrices corresponding to type-A and type-B DFSs, respectively, and $\mathbb{V}_S''$ corresponds to further SFSs of $\mathcal{C}$. Denote $\mathcal{F} = \mathcal{F}_A \cup \mathcal{F}_B \cup \mathcal{F}_S$ as the union of the set of indices for type-A DFSs, type-B DFSs, and all SFSs of $\mathcal{C}$.

The simplest way to implement decoding of the low-rate PSC is to employ the SC algorithm, which is originally proposed for PCs. For $r = 0, \ldots, n-1$, the SC algorithm decodes, with an

abuse of notation, as

$$\hat{u}_r = \begin{cases} W_{h_r} & \text{if } r \in \mathcal{F}_w \\ \sum_{s=0}^{r-1} \mathbb{V}_{i_r,s} \hat{u}_s & \text{if } r \in \mathcal{F} \setminus \mathcal{F}_w \\ \arg\max_u W_m^{(r)}(y_0^{n-1}, \hat{u}_0^{r-1}|u) & \text{if } r \notin \mathcal{F} \end{cases} \tag{17}$$

where h_r is the index of the helper symbol stored in u_r, i_r is the index of rows in $\mathbb{V}$ having the last non-zero element in column r, $W_m^{(r)}$ is the transition probability function for the t-th subchannel of the polarizing transformation. However, significantly better performance can be obtained by employing list or sequential decoding, as discussed below.

Tal-Vardy list decoding algorithm [9] tracks at most L partial vectors u_0^{r-1}. At each phase r this algorithm constructs the possible continuations u_0^r of the partial vectors, subject to (11) for $r \in \mathcal{F}$, and keeps for further processing L continuations with the highest probabilities $W_m^{(r)}(y_0^{n-1}, u_0^{r-1}|u_r)$. This can be implemented with a complexity $O(Ln \log n)$. The decoding complexity can be substantially reduced by employing a stack decoder [42]. A stack decoder stores in a priority queue, i.e., stack, the paths u_0^{r-1} of length r together with their score $M(u_0^{r-1}, y_0^{n-1})$, which can be chosen and calculated as given in [42] or as discussed below. At each iteration, the path with the highest score is retrieved from the priority queue, its valid continuations u_0^r are constructed and pushed into the priority queue together with their updated scores. Decoding terminates when a path of length n is retrieved from the priority queue. The decoder ensures that paths of a length r are retrieved at most L times. Performance and complexity of this method significantly depend on the the score function chosen. It was suggested in [39] to choose it as

$$M(u_0^{r-1}, y_0^{n-1}) = R(u_0^{r-1}, y_0^{n-1}) - E\left[R(v_0^{r-1}, y_0^{n-1})\right] \tag{18}$$

where the expectation is taken over channel outputs y_0^{n-1} obtained by transmitting v_0^{n-1} and

$$R(u_0^r, y_0^{n-1}) = \begin{cases} 0 & \text{if } r = -1 \\ R(u_0^{r-1}, y_0^{n-1}) & \text{if } (-1)^{u_r} S_m^{(r)}(u_0^{r-1}, y_0^{n-1}) \geq 0 \\ R(u_0^{r-1}, y_0^{n-1}) - \left| S_m^{(r)}(u_0^{r-1}, y_0^{n-1}) \right| & \text{if } (-1)^{u_r} S_m^{(r)}(u_0^{r-1}, y_0^{n-1}) < 0 \end{cases} \tag{19}$$

where $S_m^{(r)}(u_0^{r-1}, y_0^{n-1})$ are the modified log-likelihood ratios obtained via the min-sum recursion.

C. Proposed Design Procedure

We propose the following steps to design nested PSCs for source and channel models given in Corollary 1 with a given blocklength n, secret-key size $n - m_1 - m_2$, and a block-error probability P_B. These steps provide exact design parameter choices for nested PSCs, decided based on the simulation results over a large set of design parameters.

1) Apply the randomized PSC construction method described in Section III-A to construct PSCs with rate $\frac{n-m_1-m_2}{n}$ for a BSC(p) for a range of values in $p \in (p_A, 0.5]$.
2) Evaluate P_B of constructed PSCs with the sequential decoder in [39] (or a list decoder as in [9]) with list size L over a BSC for a range of crossover probabilities $\widetilde{p} \in (p_A, 0.5]$ to obtain the crossover probability p_c that results in the target P_B. Assign the PSC that gives the largest p_c as the low-rate PSC $\mathcal{C}$. Denote $\bar{p}$ and $\bar{p}_c$, respectively, as p and p_c values corresponding to the code $\mathcal{C}$. We remark that the design parameter p and the evaluation parameter $\widetilde{p}$ are not generally the same.
3) Using the inverse of the star operation, obtain the expected target distortion $E[q]$ averaged over all $x^n \in \mathcal{X}^n$ as $E[q] = \dfrac{\bar{p}_c - p_A}{1 - 2p_A}$.
4) Obtain the reliabilities of virtual subchannels of the polar transform by using the min-sum density evolution algorithm over a BSC$(\bar{p}_1)$, where $\bar{p}_1 = \dfrac{\bar{p} - p_A}{1 - 2p_A}$.
5) Arrange the subchannel reliabilities obtained in Step 4 in a descending order. Consecutively remove indices from the set $\mathcal{F}$, starting from the most reliable subchannels, until an average

distortion $\bar{q} = \frac{1}{n}\sum_{i=1}^{n} X_i \oplus X_{q,i}$ of at most $E[q]$ is achieved, where $\oplus$ denotes modulo-2 summation. Assign the remaining indices, i.e., the unremoved least reliable subchannel indices, as the frozen symbol indices of the high-rate code $\mathcal{C}_1$ that are denoted as $\mathcal{F}_1$.

Step 4 suggests that the design parameter $\bar{p}$ of $\mathcal{C}$ uniquely determines the design parameter $\bar{p}_1$ for $\mathcal{C}_1$. The total number of DFSs of $\mathcal{C}$ is $(t_{\mathrm{A}} + t_{\mathrm{B}})$, as defined in Section III-A. Therefore, if the difference between the rate of $\mathcal{C}_1$ and of $\mathcal{C}$, i.e., $\Delta R \triangleq H_b(q * p_A) - H_b(q)$, is larger than $\frac{t_{\mathrm{A}} + t_{\mathrm{B}}}{n}$, then $\mathcal{C}_1$ is a PC because DFSs are the most reliable frozen symbols. The difference $n\Delta R$ is larger than $(t_{\mathrm{A}} + t_{\mathrm{B}})$ for the SRAM PUF parameters we consider in the next section as ΔR increases with increasing p_A, which is consistent with our nested PSC construction where the high-rate code is a PC.

Remark 1. *This randomized nested PSC construction provides an additional degree of freedom such that the same code, designed for a given set of parameters, can be used for different P_B values and for different crossover values p_A by adapting the expected distortion level. A wide range of applications can therefore be addressed by using the same nested PSC.*

IV. Proposed Nested PSCs for PUFs

Consider the scenario where we generate a secret key S with length $n-m_1-m_2=128$ bits to use it in the advanced encryption standard (AES). Suppose intellectual property (IP) in a field-programmable gate array (FPGA) with an SRAM PUF should be protected so that the target block-error probability P_B is 10^{-6} [44]. SRAM PUF measurement channels $P_{Y|X}$ are modeled as a BSC$(p_A = 0.15)$ [32]. We apply the design procedure proposed in Section III-C for these parameters to design Codes 1 and 2, which have blocklengths 1024 and 2048 bits, respectively, and which are decoded by using sequential decoders for list sizes $L = [8, 32, 64]$.

Code 1: Consider nested PSCs with blocklength $n = 1024$ bits. First, design PSCs of rate $128/1024$ by applying Steps 1 and 2 given in Section III-C for $L = 8$ and obtain $\bar{p}$, which

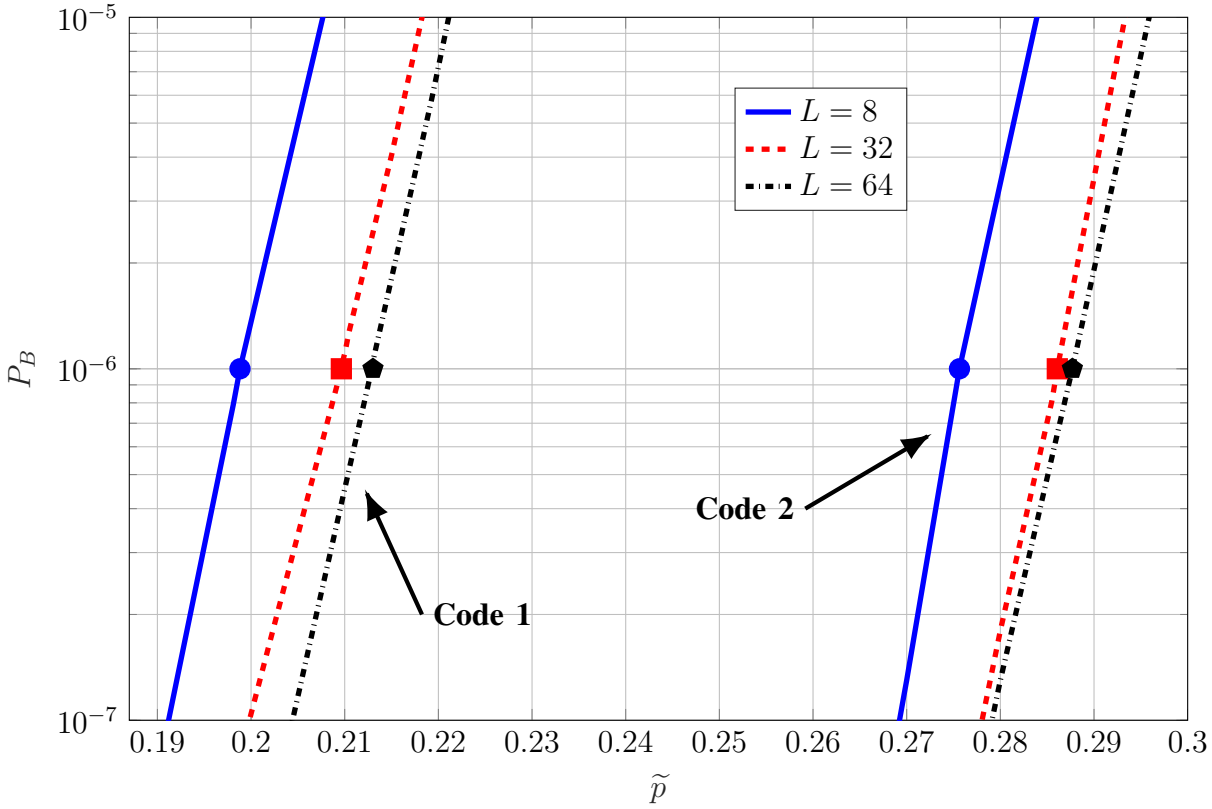

Fig. 4. Block-error probability of $\mathcal{C}$ over a BSC with crossover probabilities $\widetilde{p}$ for Codes 1 and 2 of length 1024 and 2048 bits, respectively, with sequential decoders and corresponding $\bar{p}_c$ values represented by a circle for list size $L = 8$, square for $L = 32$, and pentagon for $L = 64$.

is found to be $\bar{p} = 0.1863$. Fig. 4 depicts the $\widetilde{p}$ vs. P_B curves for the code $\mathcal{C}$ with sequential decoders for list sizes $L = [8, 32, 64]$. We observe $P_B = 10^{-6}$ in Fig. 4 at crossover probabilities of $\bar{p}_c = [0.1988, 0.2096, 0.2130]$ such that we obtain $E[q] = [0.0697, 0.0852, 0.0900]$ by Step 3 for $L = [8, 32, 64]$, respectively, where we apply $\bar{p}$ found for $L = 8$ to all list sizes for simplicity. Applying Step 4, we obtain the design parameter for the code $\mathcal{C}_1$ and evaluate the average distortion $\bar{q}$ by applying Step 5. Fig. 5 depicts the $n - m_1$ vs. $\bar{q}$ curves obtained by applying Step 5. Code 1 achieves $\bar{q} \leq E[q]$ in Fig. 5 with minimum $m_2 = [553, 492, 474]$ bits of helper data, sufficing to reconstruct a 128-bit secret key with $P_B = 10^{-6}$ for $L = [8, 32, 64]$, respectively.

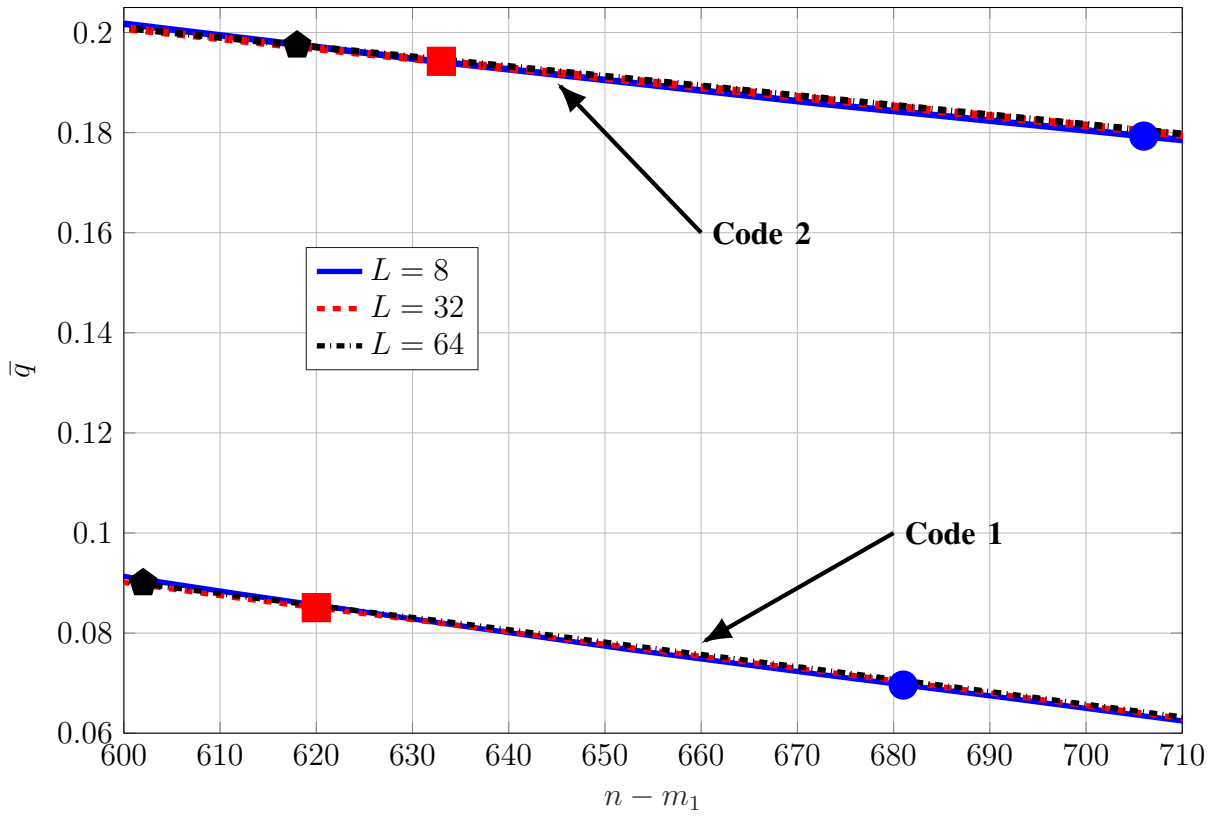

Fig. 5. Average distortion $\bar{q}$ with respect to $n - m_1$ for Codes 1 and 2 with sequential decoders and corresponding $E[q]$ values represented by a circle for list size $L = 8$, square for $L = 32$, and pentagon for $L = 64$.

Code 2: Consider nested PSCs with the same parameters as in Code 1, except $n = 2048$ bits. The value of $\bar{p}$ for this case is 0.2650. Fig. 4 shows that crossover probabilities of $\bar{p}_c = [0.2756, 0.2861, 0.2883]$ satisfy $P_B = 10^{-6}$, so the expected target distortions are $E[q] = [0.1795, 0.1944, 0.1975]$ for $L = [8, 32, 64]$, respectively. Code 2 achieves $\bar{q} \leq E[q]$ in Fig. 5 with minimum $m_2 = [578, 505, 490]$ bits, which should be stored as helper data to generate a key size of 128 bits with $P_B = 10^{-6}$ for $L = [8, 32, 64]$, respectively.

A. Rate Region Performance

We evaluate the key-leakage-storage region $\mathcal{R}_{\text{gs,binary}}$ for $p_A = 0.15$ and plot its storage-key $(R_{\text{w}}, R_{\text{s}})$ projection in Fig. 6. Furthermore, we plot in Fig. 6 the tuples achieved by Codes 1

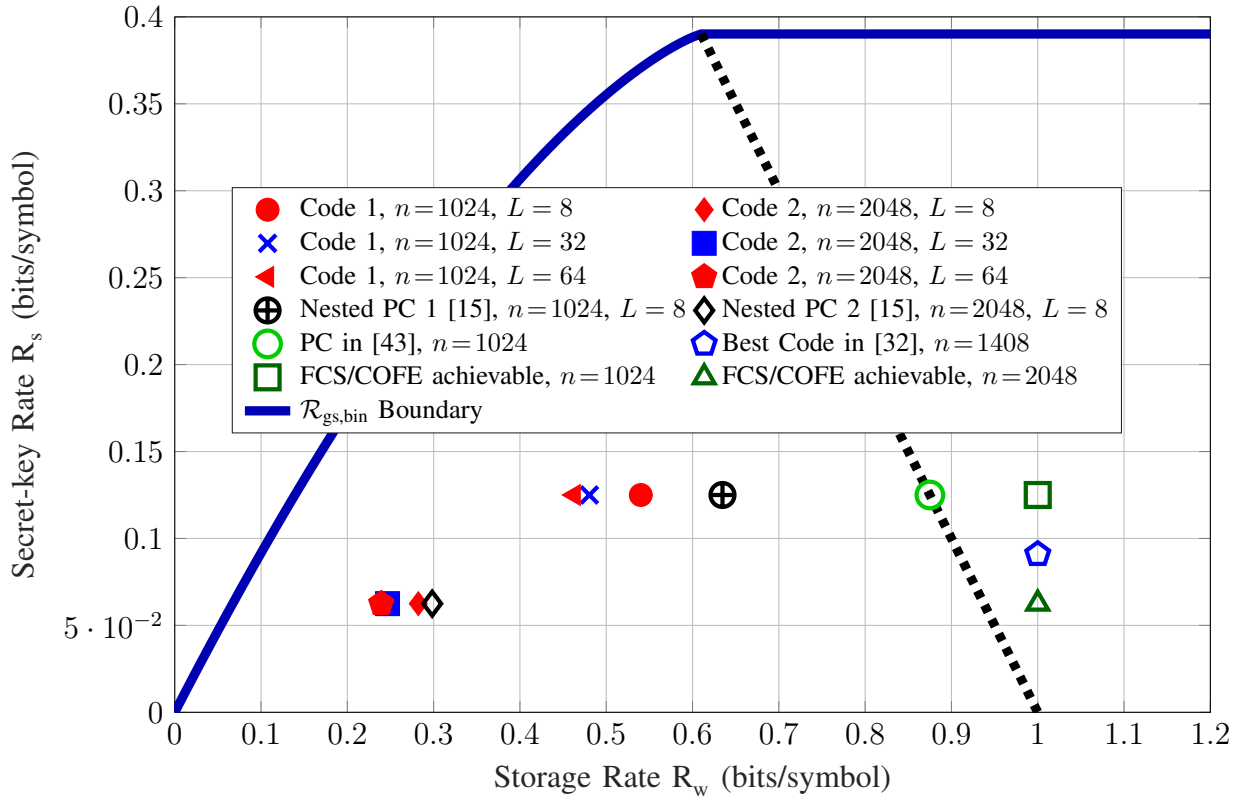

Fig. 6. Storage-key rates for key generation with crossover probability $p_A = 0.15$. The block-error probability satisfies $P_B \leq 10^{-6}$ and the key length is 128 bits for all code points. The dashed line represents $R_{\text{w}} + R_{\text{s}} = H(X) = 1$ bit/symbol. All codes with $R_{\text{w}} = 1$ bit/symbol are ECCs. The PC on the dashed line is a syndrome-coding construction.

and 2, previous nested PCs given in [15], the syndrome-coding (or SW-coding) construction proposed in [43], and the classic constructions that are code-offset fuzzy extractors (COFE) [45] and the fuzzy commitment scheme (FCS) [46].

We observe from Fig. 6 that Code 1 with $L = 8$ achieves a key vs. storage rate ratio of 0.2315, improving on the nested PC 1 ratio of 0.1969 achieved in [15] with the same list size. This result illustrates that nested PSCs achieve the best key vs. storage ratio in the literature for the same list size. Another way to compare state-of-the-art nested PCs and proposed nested PSCs is to find the minimum list size L required by nested PSCs to achieve the same storage

rate performance as being achieved by nested PCs. By following the design procedure given in Section III-C individually for all list sizes smaller than $L = 8$, we obtain that $L = 4$ suffices for nested PSCs to achieve a storage rate smaller than achieved by nested PCs for the design parameters of Code 1. We observe $P_B = 10^{-6}$ at crossover probability of $\bar{p}_c = 0.1885$, i.e., we have $E[q] = 0.0551$, for nested PSCs with $L = 4$. Code 1 with $L = 4$ achieves $\bar{q} = E[q]$ at $m_2 = 615$ bits of helper data, which is smaller than the amount 650 bits of helper data required by nested PCs with $L = 8$ for the design parameters of Code 1 [15]. Furthermore, increasing the list size of Code 1 to $L = 32$ allows to achieve a key vs. storage rate ratio of 0.2602, which is a substantial gain as compared to $L = 8$ case. A further increase in the list size does not improve the achieved ratio significantly as, e.g., Code 1 with $L = 64$ achieves 0.2698. This result might be due to the choice of the numbers t_{A} and t_{B} of type-A and type-B DFSs adapted to $L = 32$, so one might improve the performance of larger list sizes by choosing different t_{A} and t_{B}.

Code 2 with $L = 8$ achieves a $\dfrac{R_{\mathrm{s}}}{R_{\mathrm{w}}}$ ratio of 0.2215, better than 0.2095 achieved by the nested PC 2 proposed in [15]. The ratio increases to 0.2535 and 0.2612 by increasing the list size to $L = 32$ and $L = 64$, respectively. Thus, the largest $\dfrac{R_{\mathrm{s}}}{R_{\mathrm{w}}}$ ratio in the literature for SRAM PUFs is achieved by Code 1, for which we have $n = 1024$ bits, with $L = 64$. Its performance might be improved also by optimizing t_{A} and t_{B}. Furthermore, the minimum list size L required by nested PSCs to achieve the same storage rate as achieved by nested PCs with $L = 8$ for the design parameters of Code 2 is found to be $L = 6$. The block-error probability $P_B = 10^{-6}$ is achieved at crossover probability of $\bar{p}_c = 0.2720$, i.e., we have $E[q] = 0.1743$, for nested PSCs with $L = 6$. Code 2 with $L = 6$ achieves $\bar{q} = E[q]$ at $m_2 = 604$ bits of helper data, which is slightly smaller than the amount 611 bits of helper data required by nested PCs with $L = 8$ for the design parameters of Code 2 [15].

The decoding complexity of the sequential decoding algorithm in [39] depends on the quality of the measurement channel, which depends on p_A for our model. It is upper bounded by the complexity $O(Ln \log_2 n)$ of the SCL decoder, where L is the maximal number of times the

TABLE I

THE AVERAGE NUMBER OF SUMMATION AND COMPARISON OPERATIONS DONE FOR *Code 1* WITH SEQUENTIAL DECODERS.

		$\mathbf{L=4}$ $\bar{p}_c = 0.1885$	$\mathbf{L=8}$ $\bar{p}_c = 0.1988$	$\mathbf{L=32}$ $\bar{p}_c = 0.2096$	$\mathbf{L=64}$ $\bar{p}_c = 0.2130$
High-rate	**Summation Count**	12957.9	18596.4	39404.0	51431.1
PC	**Comparison Count**	9964.7	14161.5	29576.1	38358.0
Low-rate	**Summation Count**	6481.1	6512.3	6612.5	6681.3
PSC	**Comparison Count**	6152.6	6176.4	6258.3	6315.3

decoder is allowed to visit each phase (equivalent to the list size in the Tal-Vardy SCL decoding algorithm [9] used for nested PCs), but it converges to $O(n \log_2 n)$ fast with a channel bit error rate approaching 0, e.g., when $p_A \to 0$ for our model by using methods, e.g., in [29], [47], [48]. We list the average number of summation and comparison operations done with the sequential decoder of [39] for Codes 1 and 2 in Tables I and II, respectively. We remark that the low-rate PSCs are averaged over 10^8 iterations and the high-rate PCs are averaged over 20000 iterations.

Tables I and II show that increasing the list size L or the blocklength n significantly increases the decoding complexity for high-rate PCs. However, for the low-rate PSCs, increasing the list size L does not increase the decoding complexity significantly, whereas increasing the blocklength n has a similar effect on the decoding complexity as for high-rate PCs. Furthermore, low-rate PSCs have significantly lower decoding complexities than of high-rate PCs with the same L and n. Therefore, the complexity of a high-rate PC depends significantly on the list size and dominates the overall complexity of a nested PSC that is designed for the parameters chosen to protect IP in an FPGA by using SRAM PUFs.

TABLE II

THE AVERAGE NUMBER OF SUMMATION AND COMPARISON OPERATIONS DONE FOR *Code 2* WITH SEQUENTIAL DECODERS.

		$\mathbf{L=6}$ $\bar{p}_c = 0.2720$	$\mathbf{L=8}$ $\bar{p}_c = 0.2756$	$\mathbf{L=32}$ $\bar{p}_c = 0.2861$	$\mathbf{L=64}$ $\bar{p}_c = 0.2883$
High-rate	**Summation Count**	35491.2	40893.7	89803.8	108000.0
PC	**Comparison Count**	29502.2	33904.0	73957.6	88358.1
Low-rate	**Summation Count**	13825.7	13875.3	14185.9	14310.1
PSC	**Comparison Count**	13380.9	13422.4	13685.8	13791.6

V. DISCUSSIONS ON FURTHER IMPROVEMENTS

Optimal design of error-correcting PSCs is known to be a challenging task [26]. We propose in Section III-B a nested PSC construction with the aim to minimize the amount of public storage for given blocklength, low-rate code rate, channel model, (sequential) decoders, and block-error probability. Therefore, there are additional design parameters in our construction as compared to error-correcting PSC constructions and the joint effects of all design parameters on both the low-rate PSC and the high-rate PC should be revealed. We remark that our insights might differ from the insights gained from error-correcting PSC designs since we consider the joint effects of all parameters on a closely related but different problem, i.e., nested PSC design with minimum public storage rate. We next list the main insights gained from our numerous nested PSC designs, which might be useful to improve the overall nested PSC performance illustrated above.

The main parameters that can be optimized for the low-rate PSC design are, respectively, the numbers t_{A} and t_{B} of type-A and type-B DFSs, as mentioned above. We choose the parameters as in (12) and (13) that are the suggested values for error-correcting PSC designs with $L = 32$. Our simulation results suggest that these values perform well also for other list sizes considered in this work, such as $L = 8$ and 64. However, it seems to be possible to improve the overall

performance for small list sizes, which can provide a slight performance gain also for $L = 8$. Thus, more analyses are required to provide better value suggestions for t_{A} and t_{B} for nested PSC designs, especially for small list sizes. Furthermore, for all list sizes considered in Tables I and II, we use the same crossover probability value $\bar{p}$ that is obtained by applying the proposed design procedure for $L = 8$. We observe that varying $\bar{p}$ does not bring a significant overall performance gain when (12) and (13) are used to determine t_{A} and t_{B}, respectively. However, it might be the case that joint optimization of $(t_{\mathrm{A}}, t_{\mathrm{B}}, \bar{p})$ for each list size separately might result in overall performance gains.

A simple change that can reduce the hardware cost is to allow the list sizes for the low-rate PSC and high-rate PC to be different, unlike above. This additional degree of freedom enables adaptation to different hardware cost constraints imposed to enrollment and reconstruction implementations. For some applications, this change might not be possible. For instance, if a PUF is used in a mobile device for private device authentication or data encryption/decryption, then it might be preferred to use the same (sequential) decoder during enrollment and reconstruction not to increase the total hardware area required in the mobile device.

The design procedure proposed in Section III-C uses the expected target distortion $E[q]$ averaged over all realizations $x^n \in \mathcal{X}^n$ as the design parameter for the high-rate PC. However, for applications such as secret-key agreement with PUFs it is vital to provide the reliability guarantee for each PUF sequence x^n, which requires a worst-case reliability guarantee. Thus, it is illustrated in [15] that by replacing the expected target distortion $E[q]$ with the maximum distortion for, e.g., 99.99% of all realizations x^n of X^n, the same reliability guarantee can be provided with small additional public storage. Thus, the same idea can be included in the nested PSC design procedure proposed above to ensure that, e.g., 99.99% of all PUFs satisfy the reliability constraint.

VI. Conclusion

We proposed a randomized nested polar subcode construction, which can be useful for numerous information-theoretic problems. We provided a design procedure to use a polar subcode as an error-correcting code and a polar code as a vector quantizer such that the codes are nested. Nested polar subcodes are designed for the source and channel models used for SRAM PUFs to illustrate significant gains in terms of the key vs. storage rate ratio as compared to previous code designs including state-of-the-art nested polar codes. The minimum list sizes required by nested polar subcodes to perform better than state-of-the-art nested polar codes were also provided to illustrate that one can gain in hardware cost by using nested polar subcodes as compared to nested polar codes for the same design parameters due to the list size reductions. The gains from reduced list sizes were characterized for nested polar subcodes in terms of the average number of summation and comparison operations done in the sequential decoders. By analyzing simulation results and using the properties of PSCs, we proposed methods to further optimize the most promising and effective design parameters, which can lead to improvements in the privacy, reliability, secrecy, storage or hardware performance of designed nested polar subcodes. In future work, we will propose new code constructions that can perform close to the finite-length bounds one can straightforwardly calculate by combining the separate finite-length bounds for error correction and for vector quantization, which are valid also for nested code constructions considered.

Acknowledgment

O. Günlü and V. Sidorenko thank Onurcan Iscan for his insightful comments and Gerhard Kramer for improving an equation used in Section III-B1 during the Huawei 5th Professor's Day on ICT Algorithm Design in Moscow, Russia in November 2018.

REFERENCES

[1] O. Günlü, P. Trifonov, M. Kim, R. F. Schaefer, and V. Sidorenko, "Randomized nested polar subcode constructions for privacy, secrecy, and storage," in *IEEE Int. Symp. Inf. Theory Applications*, Kapolei, HI, Oct. 2020, pp. 475–479.

[2] A. D. Wyner, "The wire-tap channel," *Bell Labs Tech. J.*, vol. 54, no. 8, pp. 1355–1387, Oct. 1975.

[3] H. Mahdavifar and A. Vardy, "Achieving the secrecy capacity of wiretap channels using polar codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 10, pp. 6428–6443, Oct. 2011.

[4] M. Andersson, V. Rathi, R. Thobaben, J. Kliewer, and M. Skoglund, "Nested polar codes for wiretap and relay channels," *IEEE Commun. Lett.*, vol. 14, no. 8, pp. 752–754, Aug. 2010.

[5] O. O. Koyluoglu and H. E. Gamal, "Polar coding for secure transmission and key agreement," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 5, pp. 1472–1483, Oct. 2012.

[6] R. Liu, Y. Liang, H. V. Poor, and P. Spasojevic, "Secure nested codes for type II wiretap channels," in *IEEE Inf. Theory Workshop*, Tahoe City, CA, Sep. 2007, pp. 337–342.

[7] M. Bloch, O. Günlü, A. Yener, F. Oggier, H. V. Poor, L. Sankar, and R. F. Schaefer, "An overview of information-theoretic security and privacy: Metrics, limits and applications," *IEEE J. Sel. Areas Inf. Theory*, vol. 2, no. 1, pp. 5–22, Mar. 2021.

[8] 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, July 2009.

[9] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2213–2226, May 2015.

[10] P. W. Cuff, H. H. Permuter, and T. M. Cover, "Coordination capacity," *IEEE Trans. Inf. Theory*, vol. 56, no. 9, pp. 4181–4206, Sep. 2010.

[11] R. Ahlswede and I. Csiszár, "Common randomness in information theory and cryptography - Part I: Secret sharing," *IEEE Trans. Inf. Theory*, vol. 39, no. 4, pp. 1121–1132, July 1993.

[12] U. M. Maurer, "Secret key agreement by public discussion from common information," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 733–742, May 1993.

[13] 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.

[14] I. Csiszár and P. Narayan, "Common randomness and secret key generation with a helper," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 344–366, Mar. 2000.

[15] O. Günlü, O. İşcan, V. Sidorenko, and G. Kramer, "Code constructions for physical unclonable functions and biometric secrecy systems," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 11, pp. 2848–2858, Nov. 2019.

[16] O. Günlü and G. Kramer, "Privacy, secrecy, and storage with multiple noisy measurements of identifiers," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 11, pp. 2872–2883, Nov. 2018.

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

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

[19] O. Günlü, "Design and analysis of discrete cosine transform based ring oscillator physical unclonable functions," Master's thesis, TU Munich, Germany, Oct. 2013.

[20] T. Ignatenko, G. j. Schrijen, B. Skoric, P. Tuyls, and F. Willems, "Estimating the secrecy-rate of physical unclonable functions with the context-tree weighting method," in *IEEE Int. Symp. Inf. Theory*, Seattle, WA, July 2006, pp. 499–503.

[21] O. Günlü, "Key agreement with physical unclonable functions and biometric identifiers," Ph.D. dissertation, TU Munich, Germany, Nov. 2018, published by Dr. Hut Verlag in Feb. 2019.

[22] 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.

[23] R. A. Chou, M. R. Bloch, and E. Abbe, "Polar coding for secret-key generation," *IEEE Trans. Inf. Theory*, vol. 61, no. 11, pp. 6213–6237, Nov. 2015.

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

[25] P. Trifonov and V. Miloslavskaya, "Polar subcodes," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 2, pp. 254–266, Feb. 2016.

[26] P. Trifonov and G. Trofimiuk, "A randomized construction of polar subcodes," in *IEEE Int. Symp. Inf. Theory*, Aachen, Germany, June 2017, pp. 1863–1867.

[27] O. Günlü and O. İşcan, "DCT based ring oscillator physical unclonable functions," in *IEEE Int. Conf. Acoust., Speech Signal Process.*, Florence, Italy, May 2014, pp. 8198–8201.

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

[29] O. Günlü and R. F. Schaefer, "Low-complexity and reliable transforms for physical unclonable functions," in *IEEE Int. Conf. Acoustics, Speech, Signal Process.*, Barcelona, Spain, May 2020, pp. 2807–2811.

[30] E. Bozkir, O. Günlü, W. Fuhl, R. F. Schaefer, and E. Kasneci, "Differential privacy for eye tracking with temporal correlations," Sep. 2020, [Online]. Available: arxiv.org/abs/2002.08972.

[31] O. Günlü, K. Kittichokechai, R. F. Schaefer, and G. Caire, "Controllable identifier measurements for private authentication with secret keys," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 8, pp. 1945–1959, Aug. 2018.

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

[33] L. Kusters, O. Günlü, and F. M. Willems, "Zero secrecy leakage for multiple enrollments of physical unclonable functions," in *Symp. Inf. Theory Signal Process. Benelux*, Twente, The Netherlands, May-June 2018, pp. 119–127.

[34] G. S. Vernam, "Cipher printing telegraph systems for secret wire and radio telegraphic communications," *Trans. Amer. Inst. Elect. Eng.*, vol. 45, pp. 295–301, Jan. 1926.

[35] U. Maurer and S. Wolf, "Information-theoretic key agreement: From weak to strong secrecy for free," in *Int. Conf. Theory Applications Cryptographic Techn.*, Bruges, Belgium, May 2000, pp. 351–368.

[36] O. Günlü, T. Kernetzky, O. İşcan, V. Sidorenko, G. Kramer, and R. F. Schaefer, "Secure and reliable key agreement with physical unclonable functions," *Entropy*, vol. 20, no. 5, May 2018.

[37] 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.

[38] S. B. Korada and R. L. Urbanke, "Polar codes are optimal for lossy source coding," *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 1751–1768, Apr. 2010.

[39] P. Trifonov, "A score function for sequential decoding of polar codes," in *IEEE Int. Symp. Inf. Theory*, Vail, CO, June 2018, pp. 1470–1474.

[40] D. Kern, S. Vorkoper, and V. Kühn, "A new code construction for polar codes using min-sum density," in *Int. Symp. Turbo Codes Iterative Inf. Process.*, Bremen, Germany, Aug. 2014, pp. 228–232.

[41] A. Balatsoukas-Stimming, M. B. Parizi, and A. Burg, "LLR-based successive cancellation list decoding of polar codes," *IEEE Trans. Signal Process.*, vol. 63, no. 19, pp. 5165–5179, June 2015.

[42] K. Niu and K. Chen, "Stack decoding of polar codes," *IET Electron. Lett.*, vol. 48, no. 12, pp. 695–697, June 2012.

[43] B. Chen, T. Ignatenko, F. M. Willems, R. Maes, E. v. d. Sluis, and G. Selimis, "A robust SRAM-PUF key generation scheme based on polar codes," in *IEEE Global Commun. Conf.*, Singapore, Dec. 2017, pp. 1–6.

[44] C. Bösch, J. Guajardo, A.-R. Sadeghi, J. Shokrollahi, and P. Tuyls, "Efficient helper data key extractor on FPGAs," in *Int. Workshop Cryp. Hardware Embedded Sys.*, Washington, D.C., Aug. 2008, pp. 181–197.

[45] 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.

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

[47] L. Kusters and F. M. J. Willems, "Secret-key capacity regions for multiple enrollments with an SRAM-PUF," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 9, pp. 2276–2287, Sep. 2019.

[48] O. Günlü, R. F. Schaefer, and H. V. Poor, "Biometric and physical identifiers with correlated noise for controllable private authentication," in *IEEE Int. Symp. Inf. Theory*, Los Angeles, CA, June 2020, pp. 874–878.