MDPI

Article

Secure and Reliable Key Agreement with Physical Unclonable Functions

Onur Günlü [1,*], Tasnad Kernetzky [2], Onurcan İşcan [3], Vladimir Sidorenko [1], Gerhard Kramer [1], and Rafael F. Schaefer [4]

1 Chair of Communications Engineering, Technical University of Munich; {onur.gunlu, vladimir.sidorenko, gerhard.kramer}@tum.de
2 Associate Professorship of Line Transmission Technology, Technical University of Munich; tasnad.kernetzky@tum.de
3 Huawei Technologies Duesseldorf GmbH; onurcan.iscan@huawei.com
4 Information Theory and Applications Chair, Technische Universität Berlin; rafael.schaefer@tu-berlin.de
* Correspondence: onur.gunlu@tum.de; Tel.: +49-89-289-23470
† Parts of this paper were presented at the 2016 IEEE Global Conference on Signal and Information Processing in [1] and 2017 IEEE International Conference on Communications in [2].

Version 17 April 2018 of the paper published on 3 May 2018.

Abstract: Different transforms used in binding a secret key to correlated physical-identifier outputs are compared. Decorrelation efficiency is the metric used to determine transforms that give highly-uncorrelated outputs. Scalar quantizers are applied to transform outputs to extract uniformly distributed bit sequences to which secret keys are bound. A set of transforms that perform well in terms of the decorrelation efficiency is applied to ring oscillator (RO) outputs to improve the uniqueness and reliability of extracted bit sequences, to reduce the hardware area and information leakage about the key and RO outputs, and to maximize the secret-key length. Low-complexity error-correction codes are proposed to illustrate two complete key-binding systems with perfect secrecy, and better secret-key and privacy-leakage rates than existing methods. A reference hardware implementation is also provided to demonstrate that the transform-coding approach occupies a small hardware area.

Keywords: key agreement; physical unclonable functions; transform coding; privacy leakage; hardware implementation

1. Introduction

Secret keys stored in a device can provide intellectual property protection, and device authentication and identification. Non-volatile memory (NVM) is the traditional storage medium for secret keys. Securing the NVM is expensive due to its susceptibility to physical attacks [3]. A cheap and safe alternative to the NVM is to use physical identifiers as a source of randomness by applying the concept of *one-way functions* [4] to physical systems.

Invasive (physical) attacks to physical identifiers permanently change the identifier output so that an attacker cannot learn the secret key by using an invasive attack [4]. This property eliminates the need for continuous hardware protection [5]. Physical identifiers like physical unclonable functions (PUFs), e.g., the random start-up value of an uninitialized static random access memory (SRAM) [6] or fine variations of ring oscillator (RO) outputs [7], are considered to be random sources with high entropy [8]. Thus, we can use PUFs for low-complexity key storage in, e.g., internet of things (IoT) applications like securing a surgical robot against hacking.

There are multiple *key-generation*, or generated-secret (GS), and *key-binding*, or chosen-secret (CS), methods to reconstruct secret keys from noisy PUF outputs, where the key is generated from the PUF

outputs or bound to them, respectively. Code-offset fuzzy extractors [9] are examples of key-generation methods and the fuzzy commitment scheme [10] is a key-binding method. Code constructions based on Wyner-Ziv (WZ) coding are illustrated in [11] to asymptotically achieve the information-theoretic limits for the GS and CS models. These constructions might have high complexity, which is undesired for, e.g., IoT applications. In addition, since a key should be stored in a secure database for both models, it is more practical to allow a trusted entity to choose the secret key bound to a PUF output. Thus, in this paper, we aim at further improving reliability, privacy, secrecy, and hardware cost performance of a transform-coding algorithm, explained next, that is applied to PUF outputs in combination with the fuzzy commitment scheme.

PUFs have similar features to biometric identifiers like fingerprints. Both identifier types have correlated and noisy outputs due to surrounding environmental conditions [12]. Correlation in PUF outputs leaks information about the secret key, which causes *secrecy leakage*, and about the PUF output, causing *privacy leakage* [13–15]. Moreover, noise reduces reliability of PUF outputs and error-correction codes are needed to satisfy the reliability requirements. The transform-coding approach [16,17] in combination with a set of scalar quantizers has made its way into secret-key binding with continuous-output biometric and physical identifiers, as they allow reducing the output correlation and adjusting the effective noise at the PUF output. For instance, the discrete cosine transform (DCT) is the building block in [17] to generate a uniformly distributed bit sequence from RO outputs under varying environmental conditions. Efficient post-processing steps are applied to obtain more reliable PUF outputs rather than changing the hardware architecture, so standard components can be used. This transform-coding approach improves on the existing approaches in terms of the reliability under varying environmental conditions and maximum key length [17,18]. We apply this algorithm to PUF outputs with further significant improvements by designing the transformation and error-correction steps jointly.

Information-theoretic limits for the fuzzy commitment scheme are given in [19]. We use these information-theoretic limits to compare error-correction codes proposed for the transform-coding algorithm with the limits. Similar analyses were conducted for biometric identifiers in [20], but their assumptions such as independent and identically distributed (i.i.d.) identifier outputs and maximum block-error probability constraint $P_B = 10^{-2}$ are not realistic. We therefore consider highly correlated RO outputs with the constraint $P_B \leq 10^{-9}$, which are realistic for security applications that use PUFs [21].

1.1. Summary of Contributions and Organization

We improve the DCT-based algorithm of [17] by using different transforms and reliability metrics. We also propose error-correction codes that achieve better (secret-key, privacy-leakage) rate tuples than previous code designs. A summary of the main contributions is as follows.

- We compare a set of transforms to improve the performance of the transform coding algorithm in terms of the maximum secret-key length, decorrelation efficiency, uniqueness and security of the extracted bit sequence, and computational complexity.
- Two quantization methods with different reliability metrics are proposed to address multiple design objectives for PUFs. One method aims at maximizing the length of the bit sequence extracted from a fixed number of ROs, whereas the second method provides reliability guarantees for each output in the transform domain by fixing the decoding capability of a decoder used for error correction.
- We give a reference hardware design for the transform with the smallest computational complexity, among the set of transforms considered, in combination with the second quantization method to illustrate that our algorithm occupies a small hardware area. Our results are comparable to hardware area results of previous RO PUF designs.
- Error-correction codes that satisfy the block-error probability constraints for practical PUF systems are proposed for both quantization methods to illustrate complete key-binding systems

with perfect secrecy. The proposed codes operate at better rate tuples than previously proposed codes for the fuzzy commitment scheme. Our quantizer designs also allow us to significantly reduce the gap to the optimal (secret-key, privacy-leakage) rate point achieved by the fuzzy commitment scheme.

This paper is organized as follows. In Section 2, we define the fuzzy commitment scheme that uses PUF outputs as the randomness source. The transform-coding algorithm proposed to extract a reliable bit sequence from RO PUFs is explained in Section 3. We propose two different quantization methods with different reliability metrics in Section 4. In Section 5, we illustrate the small hardware area of the proposed algorithm with a reference hardware design, and the gains in terms of reliability, security, and maximum secret-key length as compared to the existing methods. Our proposed error-correction codes, and their secrecy and privacy performance are described in Section 6. Section 7 concludes the paper.

1.2. Notation

Upper case letters represent random variables and lower case letters their realizations. A letter with 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 the string. A random variable X has probability mass P_X or probability density f_X. Calligraphic letters such as $\mathcal{X}$ denote sets, and set sizes are denoted as $|\mathcal{X}|$. Bold letters such as $\mathbf{H}$ represent matrices. $\mathsf{Enc}(\cdot)$ is an encoder mapping and $\mathsf{Dec}(\cdot)$ is a decoder mapping. $X - Y - Z$ indicates a Markov chain. $H_b(x) = -x \log_2 x - (1-x)\log_2(1-x)$ 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 i.i.d. binary sequence of random variables with $\Pr[X_i = 1] = \alpha$ for $i = 1, 2, \dots, n$. Unif $[1 : |\mathcal{X}|]$ represents a uniform distribution over the integers from 1 to $|\mathcal{X}|$. A linear error-correction code $\mathcal{C}$ with parameters (n, k, d) has block length n, dimension k, and minimum distance d so that it can correct up to $\lfloor \frac{d-1}{2} \rfloor$ errors. .

2. System Model and the Fuzzy Commitment Scheme

Consider a RO as a source that generates a symbol $\tilde{x}$. Systematic variations in RO outputs in a two-dimensional array are less than the systematic variations in one-dimensional ROs [22]. We thus consider a two-dimensional RO array of size $L = r \times c$ and represent the array as a vector random variable $\widetilde{X}^L$. Suppose there is a single *PUF circuit*, i.e., a single two-dimensional RO array, in each device with the same circuit design, and it emits an output $\widetilde{X}^L$ according to a probability density $f_{\widetilde{X}^L}$. Each RO output is disturbed by mutually-independent additive white Gaussian noise (AWGN) and the vector noise is denoted as $\widetilde{Z}^L$. Define the noisy RO outputs as $\widetilde{Y}^L = \widetilde{X}^L + \widetilde{Z}^L$. Observe that $\widetilde{X}^L$ and $\widetilde{Y}^L$ are correlated. A secret key can thus be agreed by using these outputs of the same RO array [13,14,23,24].

One needs to extract random sequences with i.i.d. symbols from $\widetilde{X}^L$ and $\widetilde{Y}^L$ to employ available information-theoretic results for secret-key binding with identifiers. We propose an algorithm that extracts nearly i.i.d. binary and uniformly distributed random vectors $\left(X^N, Y^N\right)$ from $\widetilde{X}^L$ and $\widetilde{Y}^L$, respectively. For such X^N and Y^N, we can define a binary error vector as $E^N = X^N \oplus Y^N$. The random sequence E^N corresponds to a sequence of i.i.d. Bernoulli random variables with parameter p, i.e., $E^N \sim \text{Bern}^n(p)$. The channel $P_{Y|X}$ is thus a BSC(p).

The fuzzy commitment scheme reconstructs a secret key by using correlated random variables without leaking any information about the secret key [10]. The fuzzy commitment scheme is depicted in Fig. 1, where an encoder Enc embeds a secret key, uniformly distributed according to Unif $[1 : |\mathcal{S}|]$, into a binary codeword C^N that is added modulo-2 to the binary PUF-output sequence X^N during enrollment. The resulting sequence is the public helper data M, which is sent through an authenticated and noiseless channel. The modulo-2 sum of the helper data M and Y^N gives the result

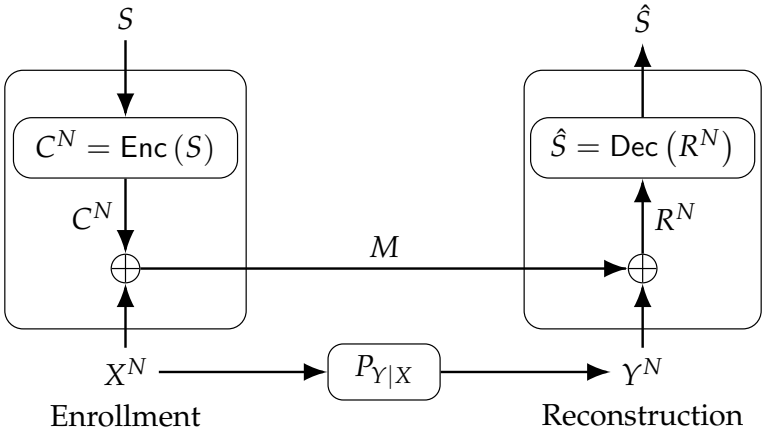

Figure 1. The fuzzy commitment scheme.

$$R^N = M \oplus Y^N = C^N \oplus E^N \tag{1}$$

which is later mapped to an estimate $\hat{S}$ of the secret key by the decoder Dec during reconstruction.

Definition 1. *A secret-key vs. privacy-leakage rate pair (R_s, R_l) is achievable by the fuzzy commitment scheme with perfect secrecy, i.e., zero secrecy leakage, if, given any $\epsilon > 0$, there is some $N \geq 1$ and an encoder and decoder for which $R_s = \dfrac{\log_2 |\mathcal{S}|}{N}$ and*

$$\Pr[S \neq \hat{S}] \leq \epsilon \qquad (\textit{reliability}) \tag{2}$$

$$I(S;M) = 0 \qquad (\textit{perfect secrecy}) \tag{3}$$

$$\frac{1}{N} I\left(X^N; M\right) \leq R_l + \epsilon \qquad (\textit{privacy}). \tag{4}$$

Theorem 1 ([19])**.** *The achievable secret-key vs. privacy-leakage rate region for the fuzzy commitment scheme with a channel $P_{Y|X}$ that is a BSC(p), uniformly distributed X and Y, and zero secrecy leakage is*

$$\mathcal{R} = \{ (R_s, R_l) : 0 \leq R_s \leq 1 - H_b(p), \quad R_l \geq 1 - R_s \}. \tag{5}$$

The region $\mathcal{R}$ suggests that any (secret-key, privacy-leakage) rate pair that sums up to 1 bit/source-bit is achievable with the constraint that the secret-key rate is at most the channel capacity of the BSC. Furthermore, smaller secret-key rates and greater privacy-leakage rates than these rates are also achievable.

The fuzzy commitment scheme is a particular realization of the CS model. The region $\mathcal{R}_{\text{cs}}$ of all achievable (secret-key, privacy-leakage) rate pairs for the CS model with a negligible secrecy-leakage rate, where a generic encoder is used to confidentially transmit an embedded secret key to a decoder that observes Y^N and the helper data M, is given in [13] as

$$\mathcal{R}_{\text{cs}} = \bigcup_{P_{U|X}} \left\{ (R_s, R_l) : \quad 0 \leq R_s \leq I(U;Y), \quad R_l \geq I(U;X) - I(U;Y) \right\} \tag{6}$$

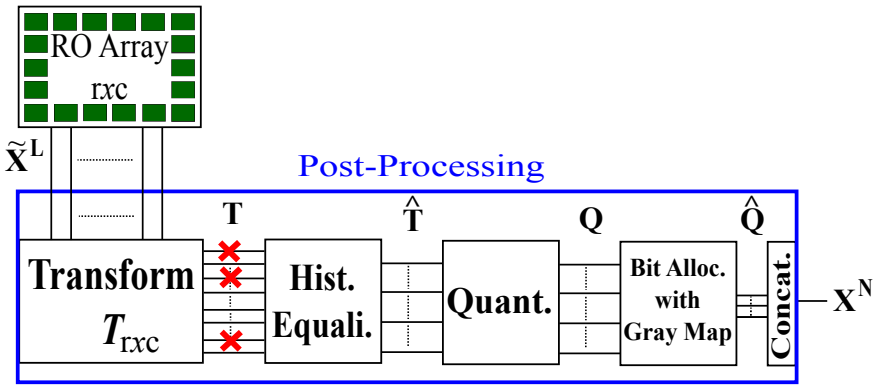

Figure 2. Transform-coding steps.

where $U - X - Y$ forms a Markov chain and the alphabet $\mathcal{U}$ of the auxiliary random variable U can be limited to have the size $|\mathcal{U}| \leq |\mathcal{X}| + 1$. The fuzzy commitment scheme is optimal, i.e., it achieves a boundary point of $\mathcal{R}_{\text{cs}}$, for a BSC $P_{Y|X}$ with crossover probability p, only at the point $(R_s^*, R_l^*) = (1 - H_b(p), H_b(p))$ [19]. This point corresponds to the highest achievable secret-key rate. Note that the region $\mathcal{R}_{\text{cs}}$ gives an outer bound for the perfect-secrecy case (see [13] for discussions).

3. Transform Coding Steps

The aim of transform coding is to reduce the correlations between RO outputs by using a linear transformation. We propose a transform-coding algorithm that extends the work in [16] and [17]. Optimizations of the quantization and error-correction parameters to maximize the security and reliability performance, and a simple method to decrease storage are its main steps. The output of these post-processing steps is a bit sequence X^N (or its noisy version Y^N) used in the fuzzy commitment scheme. We consider the same post-processing steps for the enrollment and reconstruction with the exception that during enrollment the design parameters are determined by the device manufacturer depending on the source statistics. It thus suffices to discuss only the enrollment steps. Fig. 2 shows the post-processing steps that include transformation, histogram equalization, quantization, bit assignment, and bit-sequence concatenation.

RO outputs $\widetilde{X}^L$ in an array are correlated due to, e.g., the surrounding logic [25]. A transform $T_{r\times c}(\cdot)$ of size $r \times c$ is applied to an array of RO outputs to reduce correlations. Decorrelation performance of a transform depends on the source statistics. We model each output T in the transform domain, called *transform coefficient*, obtained from a RO-output dataset in [26] by using the corrected Akaike information criterion (AICc) [27] and the Bayesian information criterion (BIC) [28]. These criteria suggest that a Gaussian distribution can be fitted to each transform coefficient T for the discrete cosine transform (DCT), discrete Walsh-Hadamard transform (DWHT), discrete Haar transform (DHT), and Karhunen-Loève transform (KLT), which are common transforms considered in the literature for image processing, digital watermarking, etc. [29]. We use maximum-likelihood estimation [30] to derive unbiased estimates for the parameters of Gaussian distributions.

The histogram equalization step in Fig. 2 converts the probability density of the i-th coefficient T_i into a standard normal distribution such that $\widehat{T}_i = \frac{T_i - \mu_i}{\sigma_i}$, where μ_i is the mean and σ_i is the standard deviation of the i-th transform coefficient for all $i = 1, 2, \ldots, L$. Quantization steps for all transform coefficients are thus the same. Without histogram equalization, we need a different quantizer for each transform coefficient. Therefore, the histogram equalization step reduces the storage for the quantization steps. Transformed and equalized coefficients $\widehat{T}_i$ are independent if the transform $T_{r\times c}(\cdot)$ decorrelates the RO outputs perfectly and the transform coefficients T_i are jointly Gaussian. One can thus use a scalar quantizer for all coefficients without a performance loss. We propose scalar quantizer and bit extraction methods that satisfy the security and reliability

requirements of the fuzzy commitment scheme with the independence assumption, in combination with a correlation-thresholding approach, as discussed below.

4. Quantizer and Code Designs

The aim of the post-processing steps in Fig. 2 is to extract a uniformly-random bit sequence X^N. We use a quantizer $Q(\cdot)$ with quantization-interval values $k = 1, 2, \cdots, 2^{K_i}$, where K_i is the number of bits we extract from the i-th coefficient $\widehat{T}_i$ for $i = 1, 2, \ldots, L$. We have

$$Q(\hat{t}_i) = k \quad \text{if} \quad b_{k-1} < \hat{t}_i \leq b_k \tag{7}$$

and we choose $b_k = \Phi^{-1}\left(\frac{k}{2^{K_i}}\right)$, where $\Phi^{-1}(\cdot)$ is the quantile function of the standard normal distribution. The quantizer output k is assigned to a bit sequence of length K_i. The chosen permutation of assigned bit sequences does not affect the security performance. However, the most likely error event when we quantize $\widehat{T}_i$ is a jump to a neighboring quantization step due to zero-mean noise. We thus apply a Gray mapping when we assign bit sequences of length K_i to the integers $k = 1, 2, \ldots, 2^{K_i}$ so that neighboring bit sequences change only in one bit position.

We next propose two different reliability metrics for joint quantizer and code designs. The first metric results in BSC measurements of each extracted bit with approximately the same crossover probability. This method extracts a different number of bits from each transform coefficient. The code design is then done for a fixed crossover probability of the BSCs. The second method fixes the maximum number of erroneous transform coefficients and considers an error-correction code that can correct all error patterns with up to a fixed number of errors.

4.1. Quantizer Design with Fixed Measurement Channels

Observe that with the quantizer in (7) and a Gray mapping, one can model the channel between a bit extracted from the enrollment outputs $\widetilde{X}^L$ and the corresponding bit extracted from the reconstruction outputs $\widetilde{Y}^L$ as a BSC with a fixed average crossover probability p_b. Our algorithm thus fixes an average crossover probability p_b such that the error-correction step in the fuzzy commitment scheme can satisfy the maximum block-error probability of 10^{-9}. The algorithm enforces that each output $\hat{t}_i$ results in an average bit error probability as close as possible to, but not greater than, p_b by adapting the number of bits $K_i(p_b)$ extracted from the i-th coefficient $\widehat{T}_i$ for all $i = 1, 2, \ldots, L$. We use the *average fractional Hamming distance* $D(K)$ between the quantization intervals assigned to the original and noisy coefficients as a metric to determine $K_i(p_b)$. Define

$$D_i(K) = \frac{1}{K} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left(\sum_{k=1}^{2^K} \Pr[Q(\hat{t}+\hat{n}) = k] \mathrm{HD}_k(\hat{t}) \right) \cdot f_{\widehat{T}_i}(\hat{t}) f_{\widehat{N}_i}(\hat{n}) \mathrm{d}\hat{t} \mathrm{d}\hat{n} \tag{8}$$

where $\mathrm{HD}_k(\hat{t})$ is the Hamming distance between the bit sequences assigned to the k-th quantization interval and to the interval $Q(\hat{t})$, and $\widehat{N}_i$ represents the Gaussian noise in the i-th coefficient after histogram equalization. We then determine $K_i(p_b)$ as the greatest number of bits K such that $D_i(K) \leq p_b$.

The first coefficient, i.e., DC coefficient, $\widehat{T}_1$ is not used since its value is a scaled version of the mean of the RO outputs in the array, which is generally known by an eavesdropper. Ambient-temperature and supply-voltage variations have a highly-linear effect on the RO outputs, so the DC coefficient is the most affected coefficient, which is another reason not to use the DC coefficient [18]. Therefore, the total number $N(p_b)$ of extracted bits from all transform coefficients for a fixed p_b is

$$N(p_b) = \sum_{i=2}^{L} K_i(p_b). \tag{9}$$

We calculate the maximum secret-key length $S_{\max}$ by using (5) for a BSC(p_b) with the maximum secret-key rate $R_s^* = 1 - H_b(p_b)$ as

$$S_{\max} = (1 - H_b(p_b)) \cdot N(p_b) \tag{10}$$

which is used to compare different transforms and to decide whether one can use an RO PUF with fixed number of ROs and p_b for secret-key binding. For instance, for the advanced encryption standard (AES), the minimum secret-key length is 128 bits. However, the rate region $\mathcal{R}$ in (5) is valid for large N. One thus needs to consider the rate loss due to a finite block length for a system design.

4.2. Quantizer Design with Fixed Number of Errors

We now propose a *conservative* approach, based on the assumption that either all bits extracted from a transform coefficient are correct or they all flip, to provide reliability guarantees. The correctness probability P_c of a transform coefficient is defined to be the probability that all bits associated with this coefficient are correct. We use this metric to determine the number of bits extracted from each coefficient such that there is an encoder and a bounded minimum distance decoder (BMDD) that satisfy the block-error probability constraint $P_B \leq 10^{-9}$. This approach results in reliability guarantees for the random-output RO arrays.

For a K-bit quantizer and the quantization boundaries b_k as in (7) for an equalized (i.e., standard) Gaussian transform coefficient $\hat{T}$, we obtain the correctness probability

$$P_c(K) = \sum_{k=0}^{2^K-1} \int_{b_k}^{b_{k+1}} \left[Q\Big(\frac{b_k - \hat{t}}{\sigma_{\hat{n}}}\Big) - Q\Big(\frac{b_{k+1} - \hat{t}}{\sigma_{\hat{n}}}\Big) \right] f_{\hat{T}}(\hat{t}) d\hat{t} \tag{11}$$

where $\sigma_{\hat{n}}^2$ is the noise variance and $f_{\hat{T}}$ is the probability density of the standard Gaussian distribution.

Suppose our channel decoder can correct all errors in up to $C_{\max}$ transform coefficients. Suppose further that coefficient errors occur independently and that the correctness probability $P_{c,i}(K)$ of the i-th coefficient $\widehat{T}_i$ for $i = 1, 2, \ldots, L$ is at least $\bar{P}_c(C_{\max})$. A sufficient condition for satisfying the block-error probability constraint $P_B \leq 10^{-9}$ is that $\bar{P}_c(C_{\max})$ satisfies the inequality

$$\sum_{c=C_{\max}+1}^{L} \binom{L}{c} (1 - \bar{P}_c(C_{\max}))^c \bar{P}_c(C_{\max})^{L-c} \leq 10^{-9}. \tag{12}$$

We thus determine the number K_i of bits extracted from the i-th transform coefficient as the maximum value K such that $P_{c,i}(K) \geq \bar{P}_c(C_{\max})$. Similar to Section 4.1, we choose $K_1 = 0$ so that the total number $N(C_{\max})$ of extracted bits is

$$N(C_{\max}) = \sum_{i=2}^{L} K_i. \tag{13}$$

In the worst case, the coefficients in error are the coefficients from which the greatest number of bits is extracted. We sort the numbers K_i of bits extracted from all coefficients in descending order such that $K'_i \geq K'_{i+1}$ for all $i = 1, 2, \ldots, L-1$. The channel decoder thus must be able to correct up to

$$e(C_{\max}) = \sum_{i=1}^{C_{\max}} K'_i \tag{14}$$

bit errors, which can be satisfied by using a block code with minimum distance $d_{\min} \geq 2e(C_{\max})+1$.

Suppose a key bound to physical identifiers in a device is used in the AES with a uniformly-distributed secret-key with a length of 128 bits. The block code used in the fuzzy commitment scheme should thus have a code length of at most $N(C_{\max})$ bits, code dimension of at least 128 bits, and minimum distance of $d_{\min} \geq 2e(C_{\max}) + 1$ for a fixed $C_{\max}$. The code rate should be as high as possible to operate close to the optimal (secret-key, privacy-leakage) rate point of the fuzzy commitment scheme. This optimization problem is hard to solve. We illustrate by an exhaustive search over a set of $C_{\max}$ values and over a selection of algebraic codes that there is a channel code that satisfies these constraints with a reliability guarantee for each extracted bit. Restricting our search to codes that admit low-complexity encoders and decoders is desired for IoT applications, for which complexity is the bottleneck.

Note that the listed conditions are conservative. For a given transform coefficient, the correctness probability can be significantly greater than the correctness threshold $\bar{P}_c(C_{\max})$. Secondly, due to Gray mapping, it is more likely that less than K_i bits are in error when the i-th coefficient is erroneous. Thirdly, it is also unlikely that the bit errors always occur in the transform coefficients from which the greatest number of bits is extracted. Therefore, even if a channel code cannot correct all error patterns with up to $e(C_{\max})$ errors, it can still be the case that the block-error probability constraint is satisfied. We illustrate such a case in the next section.

5. Performance Evaluations

Suppose the device output $\widetilde{X}^L$ is a vector random variable with the autocovariance matrix $\mathbf{C}_{\widetilde{\mathbf{X}}\widetilde{\mathbf{X}}}$. Consider RO arrays of sizes 8×8 and 16×16. Autocovariance matrix elements of such RO array outputs and noise are estimated from the dataset in [26]. We compare the DCT, DWHT, DHT, and KLT in terms of their decorrelation efficiency, maximum secret-key length, complexity, uniqueness, and security.

5.1. Decorrelation Performance

One should eliminate correlations between the RO outputs and make them independent to extract uniform bit sequences by treating each transform coefficient separately. We use the *decorrelation efficiency* η_c [31] as a decorrelation performance metric. Consider the autocovariance matrix $\mathbf{C}_{\mathbf{TT}}$ of the transform coefficients, so η_c of a transform is

$$\eta_c = 1 - \frac{\sum_{a=0}^{L}\sum_{b=0}^{L} |\mathbf{C}_{\mathbf{TT}}(a,b)| \mathbb{1}\{a \neq b\}}{\sum_{a=0}^{L}\sum_{b=0}^{L} |\mathbf{C}_{\widetilde{\mathbf{X}}\widetilde{\mathbf{X}}}(a,b)| \mathbb{1}\{a \neq b\}} \tag{15}$$

where the indicator function $\mathbb{1}\{a \neq b\}$ takes on the value 1 if $a \neq b$ and 0 otherwise. The decorrelation efficiency of the KLT is 1, which is optimal [31]. We list the average decorrelation efficiency results of other transforms in Table 1. All transforms have similar and good decorrelation efficiency performance for the RO outputs in the dataset in [26]. The DCT and DHT have the highest efficiency for 8×8 RO arrays, whereas for 16×16 RO arrays, the best transform is the DWHT. Table 1 indicates that increasing the array size improves η_c.

Table 1. The average RO output decorrelation-efficiency results.

	DCT	DWHT	DHT
η_c for 8×8	0.9978	0.9977	0.9978
η_c for 16×16	0.9987	0.9988	0.9986

5.2. Maximum Secret-key Length

The maximum number of bits extracted with the method given in Section 4.2 depends on the fixed number of transform coefficients that are in error. Moreover, the method uses a conservative metric. However, for the method given in Section 4.1, we can optimize the number of bits extracted from each coefficient to maximize the secret-key length. We therefore consider only the method in Section 4.1 for maximum key-length comparisons.

The secret key S should satisfy the length constraints of the cryptographic primitives that use it. Consider again the AES with a 128-bit secret key. We compare different transforms by calculating the maximum secret-key lengths $S_{\max}$, defined in (10), for various crossover probabilities p_b that can be obtained by applying the post-processing steps in Fig. 2. For RO array dimensions 8×8, we show $S_{\max}$ results of the considered transforms in Fig. 3. For $p_b \leq 0.05$, R_s^* is high but $N(p_b)$ is small, so $S_{\max}$ is mainly determined by $N(p_b)$, as depicted in Fig. 3. For $p_b \geq 0.07$, $N(p_b)$ is high but R_s^* mainly determines $S_{\max}$, which is small.

The DHT, DWHT, and DCT have similar $S_{\max}$ results and the KLT has worse performance than the others, which is mainly determined by the signal-to-noise ratio (SNR) in the transform domain. This illustrates that a transform's η_c performance for the estimated RO output distribution and its $S_{\max}$ performance for the estimated RO output and noise distributions can be different. We determine a crossover probability range $\mathcal{P} = [0.05, 0.07]$ such that the secret-key length of all transforms are close to their maximum and greater than 128. For a BSC with crossover probability $p \in \mathcal{P}$, we design error-correction codes such that $P_B \leq 10^{-9}$ is satisfied. The crossover probability range considered in [21] is $[0.12, 0.14]$, while 0.14 is the only value considered in [32] for the same P_B constraint. Considering a set of crossover values rather than a single value provides more flexibility in designing error-correction codes. Our crossover probability range also allows us to use higher-rate codes than the codes for the range $[0.12, 0.14]$ since the maximum key rate R_s^* of the fuzzy commitment scheme increases with decreasing p_b. The proposed transform-coding algorithm with the first quantizer method is thus beneficial for code design due to smaller crossover probability p_b.

The maximum number of extracted bits, which corresponds to N in (9), for an 8×8 RO array is 16 bits for the *1-out-of-8 masking* scheme [7], 32 bits for the *non-overlapping RO pairs* [7], and 64 bits for the *regression-based distillers* [33]. Even if one assumes no errors, i.e., $R_s^* = 1$, for these methods, their $S_{\max}$ results are much smaller than the $S_{\max}$ results of our algorithm, as shown in Fig. 3.

5.3. Transform Complexity

We measure the complexity of a transform in terms of the number of operations required to compute the transform and the hardware area required to implement it in a field-programmable gate array (FPGA). We are first interested in a computational-complexity comparison for RO arrays of sizes $r = c = 8$ and $r = c = 16$, which are powers of 2, so that fast algorithms are available for the DCT, DWHT, and DHT. We then present an RO PUF hardware design for the transform with the minimum computational complexity.

The computational complexity of the KLT for $r = c = N$ is $O(N^3)$, while it is $O(N^2 \log_2 N)$ for the DCT and DWHT, and $O(N^2)$ for the DHT [29]. There are efficient implementations of the DWHT without multiplications [34]. The DWHT is thus a good candidate for RO PUF designs for, e.g., internet of things (IoT) applications.

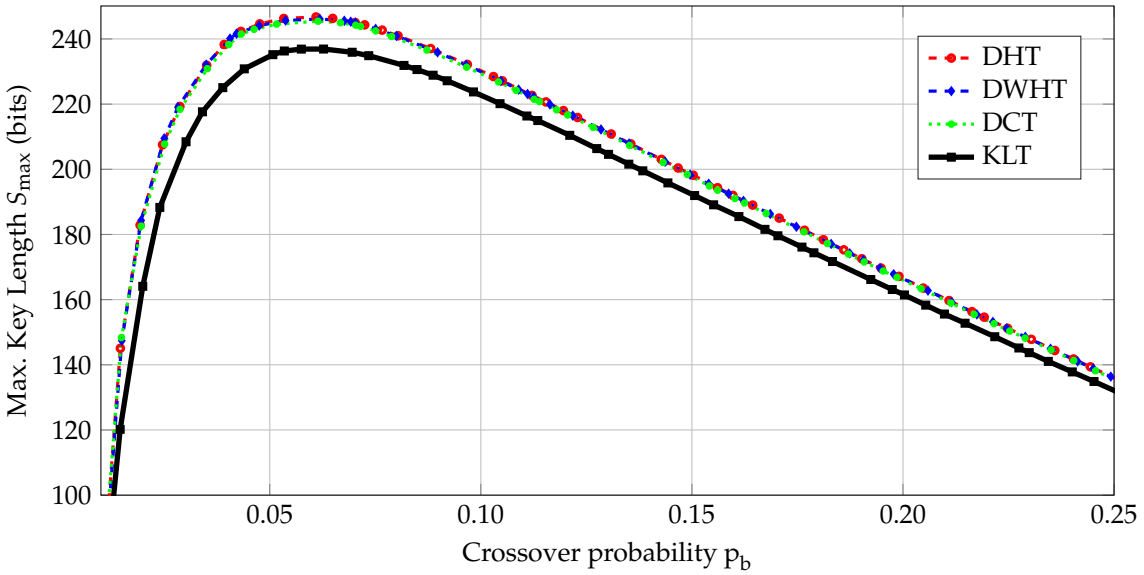

Figure 3. The maximum key lengths $S_{\max}$ for 8×8 RO arrays.

We now give a reference FPGA implementation for the DWHT without multiplications to illustrate that the hardware area occupied by the transform-coding algorithm is small and the processing time is significantly better than previous RO PUF designs.

5.3.1. FPGA Implementation

We use a Xilinx ZC706 evaluation board with a Zynq-7000 XC7Z045 system-on-chip (SoC) to evaluate our DWHT design. A high level overview of the design is depicted in Fig. 4. The Zynq SoC consists of an FPGA part and an ARM Cortex-A9 dual-core processor, connected with memory-mapped AXI4 buses [35]. The ARM processor is connected to three components: the RO array, DWHT, and quantizer. The RO array is connected via a bi-directional memory-mapped AXI bus, and the other components are connected via AXI streaming buses [36]. We first measure RO outputs with counters, give the counter values as input to the DWHT, and then quantize the transform coefficients to assign bits. This is an implementation of the transform-coding algorithm given in Fig. 2.

We use a standard RO array of size 16×16. All ROs in a row are connected to a counter and ROs in the same row can be measured serially by using the counter. There is an additional counter that stops the counting operations after a specified time. For the FPGA we use, it is practically necessary to use at least five inverters for each RO since using three inverters results in oscillation frequencies of about 1GHz, which violates the timing constraints of the FPGA. Our RO designs with five inverters operate reliably and give oscillation frequencies in the range $[400, 500]$ MHz. Furthermore, we use 16-bit counters so that the minimum duration $T_{\min}$ to have an overload in a counter is

$$T_{\min} = \frac{2^{16}-1}{500\text{MHz}} = 131\mu\text{s}. \tag{16}$$

We therefore count each RO output for a duration of 100μs, which is less than $T_{\min}$ to avoid overloads. This results in a total counting duration of 1.6ms for all 16 columns of the RO array, which is compared below with the previous RO PUF designs.

We next implement an extended version of the algorithm, proposed for an 8×8 array, in [34] to calculate the two-dimensional (2D) 16×16 DWHT without multiplications. The main block we use is the 4-point (4P)-2D DWHT [34] that takes four inputs $[x_0, x_1, x_2, x_3]$ and calculates

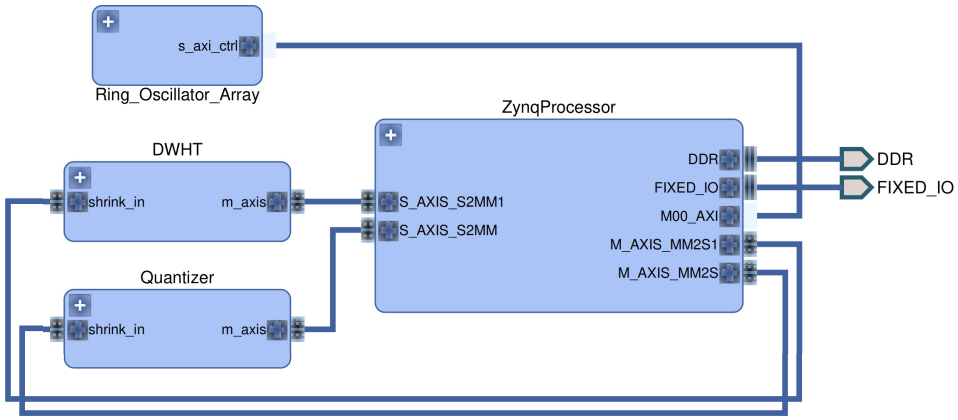

Figure 4. Hardware design overview.

$$\begin{bmatrix} y_0 & y_1 \\ y_2 & y_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} x_0 + x_1 + x_2 + x_3 & x_0 - x_1 + x_2 - x_3 \\ x_0 + x_1 - x_2 - x_3 & x_0 - x_1 - x_2 + x_3 \end{bmatrix}. \tag{17}$$

We successively apply the 4P-2D DWHT to the 16×16 RO array according to an extension of the input-selection algorithm proposed in [34]. We implement a finite state machine (FSM) to control the input and output AXI streaming interfaces as well as the input-selection algorithm. The building blocks of our DWHT implementation is depicted in Fig. 5, which includes

- a data random access memory (RAM) to store all array elements,
- a 32-bit index read-only memory (ROM), where each word stores four 8-bit array-element addresses,
- a multiplexer (MUX) to select the RAM address to be accessed,
- a second MUX to select the ROM input,
- a register for each input to convey different RAM words to different ports.

We first store all RO outputs in the data RAM. Then, the first word of the index ROM is fetched. This word holds the addresses of four array elements to be loaded. These array elements are passed to the 4P-2D DWHT's input registers by selecting the corresponding port in the address MUX and register bank. After evaluating the 4P-2D DWHT, the new array elements $[y_0, y_1, y_2, y_3]$ are written back to the locations from where the inputs $[x_0, x_1, x_2, x_3]$ were fetched. The FSM performs the same steps for all remaining ROM words and conveys the 2D DWHT coefficients to the AXI output port.

The addition and subtraction operations on four numbers in each 4P-2D DWHT evaluation requires at most two additional bits, while the subsequent bit shift to implement the division by 2 in (17) removes one bit. Since the 4P-2D DWHT is applied in total four times to each RAM location, the transform requires 20-bit operations and storage in order to process the 16-bit signed numbers used for counter values.

The quantizer contains AXI stream ports, an FSM, and one ROM. The ROM holds $2^{K_i} - 1$ quantization boundaries for the i-th transform coefficient. We remark that the histogram equalization step in Fig. 2 is useful when the number of bits K_i extracted are large, but we choose $K_i = K = 1$ for all used transform coefficients, which is illustrated in combination with an error-correction code design in Section 6.2. Therefore, we do not apply the histogram equalization step for this case, so the ROM contains 255 words and is of size 638 Bytes ($\geq 255 * 20$ bits) in total. The FSM compares the quantizer input with the corresponding quantization boundary to assign a bit 1 for transform-coefficient values greater than the quantization boundary, and the bit 0 otherwise. The assigned bits are then conveyed to the output port.

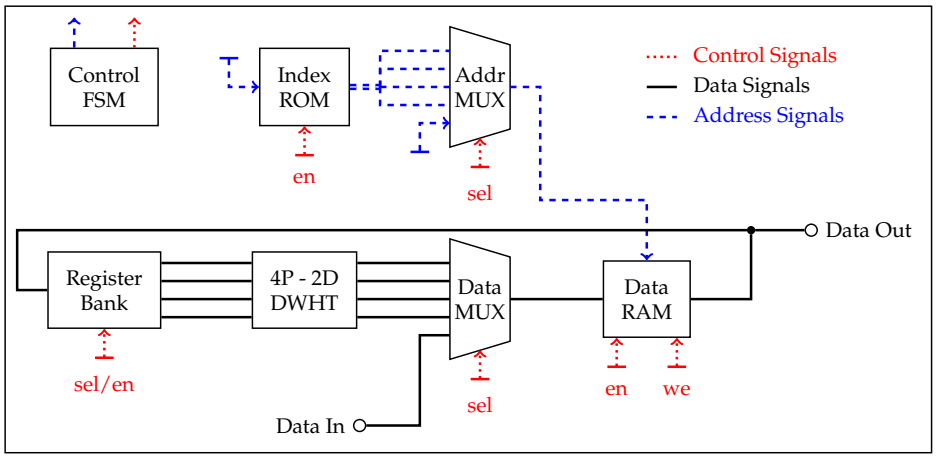

Figure 5. Building blocks for the DWHT implementation.

5.3.2. Hardware Design Comparisons

We now compare our results with another RO PUF hardware design given in [21] in terms of the hardware area and processing times. The number of LUTs, registers, and MUXs used in [21] are not available. However, our results can be compared with their slice-count and processing-delay results since the FPGA (Spartan-6) used in [21] also has 4 LUTs, 8 registers and 3 MUXes in each slice, the same as the FPGA used in this work. In addition, the quantizer and DWHT clock rate is 54MHz, as in [21]. There are alternative RO PUF designs in [37,38], but their secret-key lengths are smaller than 128 bits, which makes a comparison with our scheme difficult. Therefore, we list in Table 2 the hardware area occupied by individual components of our RO PUF design and by the RO PUF design of [21].

Table 2 illustrates that the RO array causes the highest hardware cost and uses approximately 82% of all occupied LUTs, 62% of registers, and 86% of slices. We do not include the area for RAMs and ROMs, because we use Block RAM slices that are available in the FPGA. However, we include the control logic area required to control the Block RAM slices. Our DWHT-based design occupies an approximately 11% smaller RO PUF hardware area than the RO PUF design proposed in [21] in terms of the number of slices used. This result can be improved if we re-use the same area for different ROs, which might increase correlations in the RO outputs. In addition, the DWHT and quantizer constitute approximately 14% of the total slice count of our RO PUF design. These results illustrate that the transform-coding approach occupies a small hardware area.

The total counter duration of 1.6ms is a result of the calculation given in (16) to avoid overloads in the counters, and the choice of this value depends mainly on the number of inverters used for each RO and counter bit width. The overall processing time of the proposed design is approximately 1.68ms, which is significantly better than the processing delay of the RO PUF design in [21].

5.4. Uniqueness and Security

The bit sequence extracted from a physical identifier should consist of uniformly distributed bits so that the rate region $\mathcal{R}$ in (5) is valid. A common measure, called *uniqueness*, for checking randomness of a bit sequence is the average fractional Hamming distance between the bit sequences extracted from different RO PUFs [17]. We obtain similar uniqueness results for all transforms, where the mean Hamming distance is 0.500 and Hamming distance variance is approximately 7×10^{-4}. All transforms thus provide close to optimal uniqueness results due to their high decorrelation efficiencies and equipartitioned quantization intervals. These results are significantly better than the results 0.462 [7] and 0.473 [26].

The National Institute of Standards and Technology (NIST) provides a set of randomness tests that check whether a bit sequence can be differentiated from a uniformly random bit sequence [39].

Table 2. Hardware area and processing delays for RO PUF designs.

Blocks	**LUTs**	**Registers**	**MUXes**	**RAM&ROM** [Byte]	**Slices**	**Duration** [μs]
Proposed-ROs	1632	397	65	0	729	1600
Proposed-DWHT	326	200	0	1664	99	66
Proposed-Quantizer	43	39	0	638	21	14
Proposed (ROPUF)	2001	636	65	2302	849	1680
PUFKY (ROPUF) [21]	n.a.	n.a.	n.a.	n.a.	952	4611

We apply these tests to evaluate the randomness of the generated sequences. We observe that the bit sequences generated from ROs in the dataset [26] with the DWHT pass most of the applicable tests for short lengths for both reliability metrics, which is considered to be an acceptable result [39]. We also conclude that the KLT performs the best due to its optimal decorrelation performance. One can apply a thresholding approach such that the reliable transform coefficients from which the bits are extracted do not have high correlations, which further improves the security performance [18].

6. Privacy and Secrecy Analysis of Proposed Error-correction Codes

Suppose that extracted bit sequences are uniformly distributed so that the secrecy leakage is zero. We propose different codes for the transform-coding algorithm according to the two proposed reliability metrics.

6.1. Codes for the Quantizer Design with Fixed Measurement Channels

For the first quantizer method given in Section 4.1, fix an average crossover probability $p_b = 0.06$ to obtain the highest maximum secret-key length, as shown in Fig. 3. We illustrate that there are efficient error-correction codes for the fuzzy commitment scheme with $P_B \leq 10^{-9}$ and a small privacy-leakage rate. Recall that the code dimension has to be at least 128 bits, a requirement of the AES, so the block length is in the short block-length regime for error-correction codes with high rates and $k = 128$. We expect a rate loss in our code designs due to the small block-error probability constraint and short block length. One needs finite-length bounds for the fuzzy commitment scheme, which are not available in the literature. We thus compare the performance of our codes with the region $\mathcal{R}$ given in (5). The basic approach to design codes for small block-error probabilities and reasonable decoding complexity is to use concatenated codes. Since the hardware complexity of a code design should be small for IoT applications, we minimize also the field sizes of the codes.

Remark 1. It would be natural to use iterative decoders in combination with high-performance codes like low density parity check (LDPC) and turbo codes. However, hardware complexity might increase and it is a difficult task to simulate these codes for $P_B \leq 10^{-9}$. We thus use concatenated algebraic codes so that we can find analytical bounds on P_B without simulations for the outer code.

The first construction uses a Reed-Muller (RM) code $\mathcal{C}(32, 6, 16)$ as the inner code and a Reed-Solomon (RS) code $\mathcal{C}(28, 22, 7)$ that operates with symbols from the Galois field $\mathbb{F}_{2^6}$ as the outer code of a concatenated code. Every symbol of the RS code can be represented by 6 bits and the code takes 22 symbols as input, which corresponds to 132 input bits that is greater than 128 bits. The majority logic decoder (MLD) of the inner RM code transforms the BSC with crossover probability $p_b = 0.06$ into a channel with errors and erasures by declaring an *erasure* if there are two codewords with equal distances to a received vector and makes an *error* if a wrong codeword is selected. Simulation results show that the erasure probability after the MLD of the inner code is about 6.57×10^{-5} and the error probability is about 4.54×10^{-6}. The BMDD for the outer code correctly reconstructs the codeword if $2 \cdot e + \nu < d$, where e is the number of errors and ν is the number of erasures in the received vector [40]. The block-error probability after decoding the outer RS code is approximately $P_B \approx 1.37 \times 10^{-11}$. The key and leakage rates of this code are $R_s = 0.1473$ and $R_l = 0.8527$ bits/source-bit, respectively.

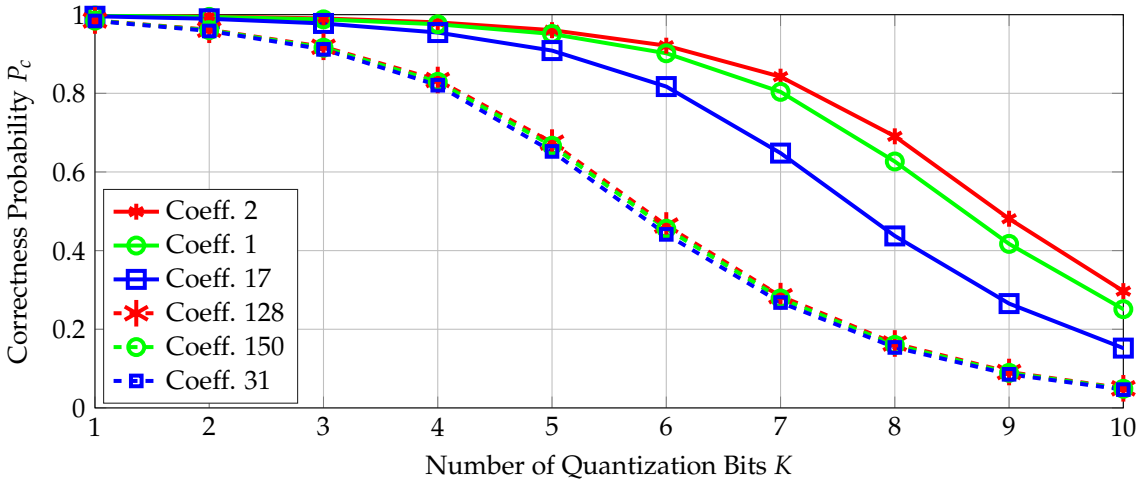

Figure 6. The correctness probabilities for transform coefficients.

An alternative concatenated code is a binary extended Bose-Chaudhuri-Hocquenghem (BCH) code $\mathcal{C}(256, 132, 36)$ as the outer code and a repetition code $\mathcal{C}(3, 1, 3)$ as the inner code. The maximum-likelihood decoder for the inner code transforms the BSC with crossover probability $p_b = 0.06$ into a BSC with $p_b = 0.0104$ so that the BMDD for the outer BCH code results in $P_B = 3.48 \times 10^{-10}$. The key-leakage rate pair (R_s, R_l) for this code is $(0.1719, 0.8281)$ bits/source-bit, which gives better rates than the RM+RS concatenation above and the best generalized-concatenated-code (GCC) design with the fuzzy commitment scheme in [32] with the key-leakage rate pair $(0.1260, 0.8740)$ bits/source-bit, which is shown to be better than the previous results in [21]. The significant improvement in the rates with a low-complexity code is due to the decrease in p_b by using our transform-coding algorithm.

The fuzzy commitment scheme can asymptotically achieve the maximum secret-key rate $R_s^* = 0.6726$ bits/source-bit and corresponding minimum privacy-leakage rate $R_l^* = 0.3274$ bits/source-bit for a BSC$(p_b = 0.06)$. Better key-leakage rate pairs are thus possible, e.g., by using GCCs or by improving the decoder for the outer code. However, these constructions would result in increased hardware complexity, which is not desired for IoT applications.

6.2. Codes for the Quantizer Design with Fixed Number of Errors

We now select a channel code according to Section 4.2 to store a secret key of length 128 bits. The correctness probabilities defined in (11) for the transform coefficients T with the three highest and three smallest probabilities are plotted in Fig. 6. The indices of the 16×16 transform coefficients follow the order in the dataset [26], where the coefficient index at the first row and first column is 1, and it increases columnwise up to 16 so that the second row starts with the index 17, the third row with the index 33, etc. The most reliable transform coefficients are the low-frequency coefficients, which are in our case at the upper-left corner of the 2D transform-coefficient array with indices such as $1, 2, 3, 17, 18, 19, 33, 34, 35$. The low-frequency transform coefficients therefore have the highest SNRs for the source and noise statistics obtained from the RO dataset in [26]. The least reliable coefficients are observed to be spatially away from the transform coefficients at the upper-left or lower-right corners of the 2D transform-coefficient array. These results indicate that the *SNR-packing efficiency*, which can be defined similarly as the energy-packing efficiency, of a transform follows a more complicated scan order than the classic zig-zag scan order used for the energy-packing efficiency metric [41]. Observe from Fig. 6 that increasing the number of extracted bits decreases the correctness probability for all coefficients since the quantization boundaries get closer so that errors due to noise become more likely, i.e., the probability $P_c(K)$ defined in (11) decreases with increasing K.

Table 3. Code-parameter constraints.

$C_{\max}$	16	17	18	19	20
$\bar{P}_c$	0.9902	0.9889	0.9875	0.9860	0.9844
$K_{\max}$	3	3	3	3	3
N	144	224	250	255	259
e	18	20	21	23	25

We fix the maximum number $C_{\max}$ of transform coefficients T allowed to be in error and calculate the correctness threshold $\bar{P}_c(C_{\max})$ using (12), the total number $N(C_{\max})$ of extracted bits using (13), and the number $e(C_{\max})$ of errors the block code should be able to correct using (14). We observe that if $C_{\max} \leq 10$, $\bar{P}_c(C_{\max})$ is so large that $P_{c,i}(K=1) \leq \bar{P}_c(C_{\max})$ for all $i = 2, \ldots, L$. If $11 \leq C_{\max} \leq 15$, $N(C_{\max})$ is less than the required code dimension of 128 bits. Increasing $C_{\max}$ results in a smaller correctness threshold $\bar{P}_c(C_{\max})$ so that the maximum of the number $K_{\max}(C_{\max}) = K_1'(C_{\max})$ of bits extracted among the $L-1$ used coefficients increases. This approach can increase hardware complexity. We thus do not consider the cases where $C_{\max} > 20$. Table 3 shows $\bar{P}_c(C_{\max})$, $N(C_{\max})$, and $e(C_{\max})$ for the remaining range of $C_{\max}$ values, which are used for channel-code selection.

Consider again binary (extended) BCH and RS codes, which have good minimum-distance properties. An exhaustive search does not provide a code with dimension of at least 128 bits and with parameters satisfying any of the $(N(C_{\max}), e(C_{\max}))$ pairs in Table 3. However, the correctness threshold analysis leading to Table 3 is conservative. We therefore choose a BCH code with parameters as close as possible to a $(N(C_{\max}), e(C_{\max}))$ pair and then prove that even if the number e_{BCH} of errors the chosen BCH code can correct is less than $e(C_{\max})$, the block-error probability constraint is satisfied. Consider therefore the BCH code with the block length 255, code dimension 131, and a capability of correcting all error patterns with $e_{\text{BCH}} = 18$ or less errors.

We now show that the proposed code satisfies the block-error probability constraint. First, we impose the condition that exactly one bit is extracted from each coefficient, i.e., $K_i = 1$ for all $i = 2, 3, \ldots, L$, so that in total $N = L - 1 = 255$ bits are obtained. Note that this results in independent bit errors E_i. It follows from this condition that the chosen block code should be able to correct all error patterns with up to $e = 20$ bit errors rather than $e(20) = 25$ bit errors, which is still greater than the error-correction capability $e_{\text{BCH}} = 18$ of the considered BCH code.

The block error probability P_B for the BCH code $\mathcal{C}(255, 131, 37)$ with a BMDD corresponds to the probability of having more than 18 errors in the codeword, i.e.,

$$P_B = \sum_{j=19}^{255} \left[\sum_{A \in \mathcal{F}_j} \prod_{i \in A} (1 - P_{c,i}) \cdot \prod_{i \in A^c} P_{c,i} \right] \tag{18}$$

where $P_{c,i}$ is the correctness probability of the i-th transform coefficient $\widehat{T}_i$ defined in (11) for $i = 2, 3, \ldots, 256$, $\mathcal{F}_j$ is the set of all size-j subsets of the set $\{2, 3, \ldots, 256\}$, and A^c denotes the complement of the set A. The correctness probabilities $P_{c,i}$ are different and they represent probabilities of independent events due to the independence assumption for the transform coefficients.

One needs to consider $\sum_{j=0}^{18} \binom{255}{j} \approx 1.90 \times 10^{27}$ different cases to calculate (18), which is not practical. We thus use the discrete Fourier transform - characteristic function (DFT-CF) method [42] to calculate the block-error probability and obtain the result $P_B \approx 1.26 \times 10^{-11} < 10^{-9}$. The block-error probability constraint is thus satisfied by using the BCH code $\mathcal{C}(255, 131, 37)$ with a BMDD although the conservative analysis suggests that it would not be satisfied.

We now compare the BCH code $\mathcal{C}(255, 131, 37)$ with previous codes proposed for binding keys to physical identifiers with the fuzzy commitment scheme and a secret-key length of 128 bits such that $P_B \leq 10^{-9}$ is satisfied. The (secret-key, privacy-leakage) rate pair for this proposed code is $(R_s, R_l) = (\frac{131}{255}, 1 - \frac{131}{255}) \approx (0.514, 0.486)$ bits/source-bit. This pair is significantly better than our

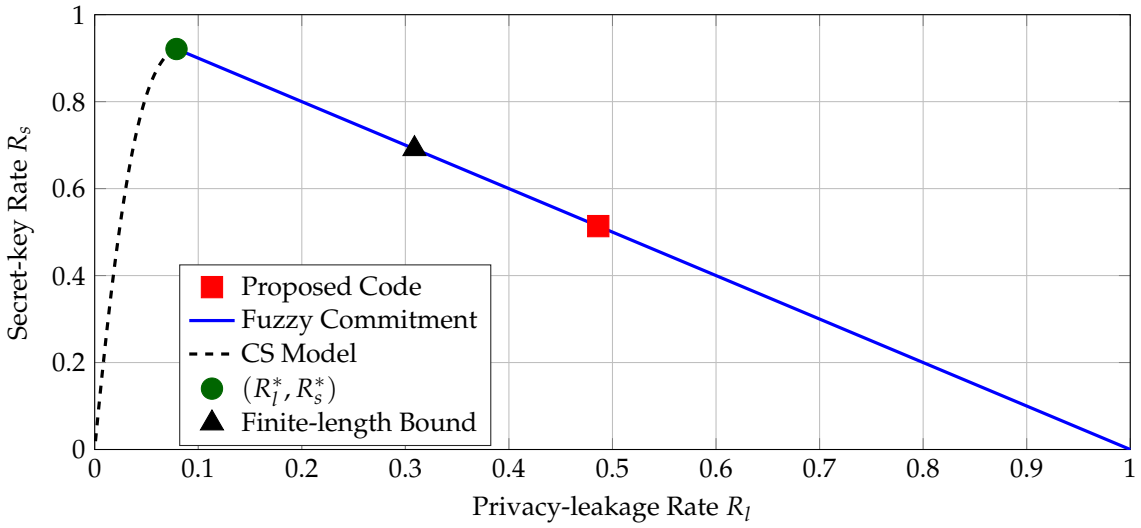

Figure 7. The operation point of the proposed BCH code $\mathcal{C}(255, 131, 37)$, regions of achievable rate pairs according to (5) and (6), the maximum secret-key rate point, and a finite-length bound for $N = 255$ bits, $P_B = 10^{-9}$, and BSC(0.0097).

previous results in Section 6.1 proposed for a BSC($p_b = 0.06$). The main reason for obtaining a better (secret-key, privacy-leakage) rate pair is that the quantizer in Section 4.2 allows us to exploit higher identifier-output reliability by decreasing the number of bits extracted from each transform coefficient.

We compare the secret-key and privacy-leakage rates of the BCH code $\mathcal{C}(255, 131, 37)$ with the region of all achievable rate pairs for the CS model and the fuzzy commitment scheme for a BSC $P_{Y|X}$ with crossover probability $p_b = 1 - \frac{1}{L-1}\sum_{i=2}^{L} P_{c,i}(K_i = 1) \approx 0.0097$, i.e., the probability of being in error averaged over all used transform coefficients with the quantizer in Section 4.2. We compute the boundary points of the region $\mathcal{R}_{\text{cs}}$ by using Mrs. Gerber's lemma [43], which gives the optimal auxiliary random variable U in (6) when $P_{Y|X}$ is a BSC. We plot the regions of all rate pairs achievable with the fuzzy commitment scheme and CS model, the maximum secret-key rate point, the (secret-key, privacy-leakage) rate pair of the proposed code, and a finite-length bound [44] for the block length of $N = 255$ bits and $P_B = 10^{-9}$ in Fig. 7.

The maximum secret-key rate is $R_s^* \approx 0.922$ bits/source-bit with a corresponding minimum privacy-leakage rate of $R_l^* \approx 0.079$ bits/source-bit. There is a gap between the secret-key rate of the proposed code and the only operation point where the fuzzy commitment scheme is optimal. Part of this rate loss can be explained by the short block length of the code and the small block-error probability constraint. The finite-length bound given in [44, Theorem 52] establishes that the rate pair $(R_s, R_l) = (0.691, 0.309)$ bits/source-bit is achievable by using the fuzzy commitment scheme, as depicted in Fig. 7. One can therefore further improve the rate pairs by using better codes and decoders with higher hardware complexity, but this may not be possible for IoT applications. Fig. 7 also illustrates that there exist other code constructions, e.g., the WZ-coding construction in [11], that reduce the privacy-leakage rate for a fixed secret-key rate.

7. Conclusion

The reliability, uniqueness, security, computational-complexity, and key-length performance of various transforms was compared to select the best transforms for reliable secret-key binding for RO PUFs by using the fuzzy commitment scheme. The DWHT and DHT perform best in terms of computational-complexity, maximum key length, and reliability. All transforms give close to optimal uniqueness and good security results. A reference hardware design with the DWHT showed that the hardware area required by the transform-coding approach is small and less than required by

the existing RO PUF designs. Low-complexity concatenated codes with high secret-key and small privacy-leakage rates, which are better than previous results, are proposed for a realistic block-error probability of 10^{-9}.

We further improved the transform-coding algorithm applied to physical identifiers by designing quantizers with reliability guarantees. This alternative quantizer converts the block-error probability constraint $P_B \leq 10^{-9}$ into a constraint on the number of transform coefficients allowed to be in error. We proposed a BCH code $\mathcal{C}(255, 131, 37)$ with a higher code rate than our previously proposed codes. Comparisons with the region of all achievable (secret-key, privacy-leakage) rate pairs for the fuzzy commitment scheme show that there is still a gap between the optimal rate pairs and the proposed code. This gap can be closed by using other channel codes and decoders at the cost of higher hardware complexity or by designing codes for other CS model constructions. In future work, we will apply an extension of water-filling techniques to the transform-coefficients in order to improve the reliability and security performance.

Acknowledgments: O. Günlü thanks Anes Belkacem and Bernhard C. Geiger for their contributions to one of the conference papers used in this work. 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. G. Kramer was supported by an Alexander von Humboldt Professorship endowed by the German Federal Ministry of Education and Research.

References

1. Günlü, O.; İşcan, O.; Sidorenko, V.; Kramer, G. Reliable secret-key binding for physical unclonable functions with transform coding. IEEE Global Conf. Sign. and Inf. Process.; Greater Washington, DC, Dec. 2016; pp. 986–991.
2. Günlü, O.; Belkacem, A.; Geiger, B.C. Secret-key binding to physical identifiers with reliability guarantees. IEEE Int. Conf. Commun.; Paris, France, May 2017; pp. 1–6.
3. Suh, G.E.; Clarke, D.; Gassend, B.; Dijk, M.V.; Devadas, S. AEGIS: Architecture for tamper-evident and tamper-resistant processing. ACM 17th Annu. Int. Conf. Supercomputing; New York, NY, June 2003; pp. 160–171.
4. Pappu, R. Physical one-way functions. PhD thesis, M.I.T., Cambridge, MA, 2001.
5. Böhm, C.; Hofer, M. *Physical unclonable functions in theory and practice*; Springer: New York, NY, 2013.
6. Guajardo, J.; Kumar, S.S.; Schrijen, G.J.; Tuyls, P. FPGA intrinsic PUFs and their use for IP protection. In *Int. Workshop Cryptographic Hardware and Embedded Systems*; Paillier, P.; Verbauwhede, I., Eds.; Berlin Heidelberg, Germany: Springer-Verlag, 2007; pp. 63–80.
7. Suh, G.E.; Devadas, S. Physical unclonable functions for device authentication and secret key generation. ACM/IEEE Design Automation Conf.; San Diego, CA, June 2007; pp. 29–14.
8. Gassend, B. Physical random functions. Master's thesis, M.I.T., Cambridge, MA, 2003.
9. Dodis, Y.; Ostrovsky, R.; Reyzin, L.; Smith, A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *Soc. Industrial Appl. Math. J. Comp.* Mar. **2008**, *38*, 97–139.
10. Juels, A.; Wattenberg, M. A fuzzy commitment scheme. ACM Conf. Comp. and Commun. Security; New York, NY, Nov. 1999; pp. 28–36.
11. Günlü, O.; İşcan, O.; Sidorenko, V.; Kramer, G. Wyner-Ziv coding for physical unclonable functions and biometric secrecy systems. Sep. 2017, [Online]. Available: arxiv.org/abs/1709.00275.
12. Maes, R. *Physically unclonable functions*; Berlin-Heidelberg, Germany: Springer-Verlag, 2013.
13. Ignatenko, T.; Willems, F. Biometric systems: Privacy and secrecy aspects. *IEEE Trans. Inf. Forensics and Sec.* Dec. **2009**, *4*, 956–973.
14. Lai, L.; Ho, S.W.; Poor, H.V. Privacy-security trade-offs in biometric security systems - Part I: Single use case. *IEEE Trans. Inf. Forensics and Sec.* Mar. **2011**, *6*, 122–139.
15. Günlü, O.; Kramer, G. Privacy, secrecy, and storage with noisy identifiers. Jan. 2016, [Online]. Available: arxiv.org/abs/1601.06756.
16. Günlü, O. Design and analysis of discrete cosine transform based ring oscillator physical unclonable functions. Master's thesis, Techn. Univ. Munich, Munich, Germany, 2013.

17. Günlü, O.; İşcan, O. DCT based ring oscillator physical unclonable functions. IEEE Int. Conf. Acoustics, Speech and Sign. Proc.; Florence, Italy, May 2014; pp. 8198–8201.
18. Günlü, O.; İşcan, O.; Kramer, G. Reliable secret key generation from physical unclonable functions under varying environmental conditions. IEEE Int. Workshop Inf. Forensics and Security; Rome, Italy, Nov. 2015; pp. 1–6.
19. Ignatenko, T.; Willems, F.M. Information leakage in fuzzy commitment schemes. *IEEE Trans. Inf. Forensics and Sec.* June **2010**, *5*, 2337–348.
20. Ignatenko, T.; Willems, F.M. Privacy-leakage codes for biometric authentication systems. IEEE Int. Conf. Acoustics, Speech and Sign. Proc.; Florence, Italy, May 2014; pp. 1601–1605.
21. Maes, R.; Herrewege, A.V.; Verbauwhede, I. PUFKY: A fully functional PUF-based cryptographic key generator. In *Cryptographic Hardware and Embedded Sys.*; Berlin Heidelberg, Germany: Springer-Verlag, Sep. 2012; pp. 302–319.
22. Maiti, A.; Schaumont, P. Improved ring oscillator PUF: an FPGA-friendly secure primitive. *J. Cryptology* Apr. **2011**, *24*, 375–397.
23. Ahlswede, R.; Csiszár, I. Common randomness in information theory and cryptography - Part I: Secret sharing. *IEEE Trans. Inf. Theory* July **1993**, *39*, 1121–1132.
24. Maurer, U. Secret key agreement by public discussion from common information. *IEEE Trans. Inf. Theory* May **1993**, *39*, 2733–742.
25. Eiroa, S.; Baturone, I. An analysis of ring oscillator PUF behavior on FPGAs. IEEE Int. Conf. Field-Program. Techn.; New Delhi, India, Dec. 2011; pp. 1–4.
26. Maiti, A.; others. A large scale characterization of RO-PUF. IEEE Int. Symp. Hardware-Orient. Sec. and Trust; Anaheim, CA, June 2010; pp. 94–99.
27. Sugiura, N. Further analysis of the data by Akaike's information criterion and the finite corrections. *Commun. Statistics, Theory and Methods* Jan. **1978**, *7*, 13–26.
28. Schwarz, G. Estimating the dimension of a model. *The Annals of Stat.* **1978**, *6*, 461–464.
29. Wang, R. *Introduction to orthogonal transforms: with applications in data processing and analysis*; Cambridge University Press, 2012.
30. Bishop, C.M. *Pattern recognition and machine learning*; Vol. 1, New York: Springer-Verlag, 2006.
31. Ohm, J.R. *Multimedia signal coding and transmission*; Berlin Heidelberg, Germany: Springer-Verlag, 2015.
32. Puchinger, S.; others. On error correction for physical unclonable functions. VDE Int. ITG Conf. Systems, Comm. and Coding; Hamburg, Germany, Feb. 2015; pp. 1–6.
33. Yin, C.E.; Qu, G. Improving PUF security with regression-based distiller. ACM/IEEE Design Automation Conf.; Austin, TX, May 2013; pp. 1–6.
34. Komatsu, K.; Sezaki, K. Lossless 2D discrete Walsh-Hadamard transform. IEEE Int. Conf. Acoustics, Speech and Sign. Proc.; Salt Lake City, UT, May 2001; Vol. 3, pp. 1917–1920.
35. AMBA AXI and ACE Protocol Specification AXI3, AXI4, AXI5, ACE and ACE5. Dec. 2017, [Online]. Available: developer.arm.com/docs/ihi0022/latest/amba-axi-and-ace-protocol-specification-axi3-axi4-axi5-ace-and-ace5.
36. AMBA AXI4-Stream Protocol Specification v1.0. Mar. 2010, [Online]. Available: https://developer.arm.com/docs/ihi0051/latest/amba-axi4-stream-protocol-specification-v10.
37. Sahoo, D.P.; Mukhopadhyay, D.; Chakraborty, R.S. Design of low area-overhead ring oscillator PUF with large challenge space. Int. Conf. Reconfigurable Computing FPGAs, Cancun, Mexico, Dec. 2013; pp. 9–11. pp. 1–6.
38. Parrilla, L.; Castillo, E.; Morales, D.P.; García, A. Hardware activation by means of PUFs and elliptic curve cryptography in field-programmable devices. *Electronics* Jan. **2016**, *5*.
39. Rukhin, A.; others. A statistical test suite for random and pseudorandom number generators for cryptographic applications. Technical report, National Inst. Stand. and Techno., 2001. Rev. in 2010.
40. Lin, S.; Costello, D.J. *Error control coding*; Englewood Cliffs, NJ: Prentice-Hall, 2004.
41. Chen, W.H.; Pratt, W. Scene adaptive coder. *IEEE Trans. Commun.* Mar. **1984**, *32*, 225–232.
42. Hong, Y. On computing the distribution function for the sum of independent and nonidentical random indicators. Technical report, Dep. Stat., Virginia Tech., Blacksburg, VA, Apr. 2011.
43. Wyner, A.D.; Ziv, J. A theorem on the entropy of certain binary sequences and applications: Part I. *IEEE Trans. Inf. Theory* Nov. **1973**, *19*, 769–772.

642 44. Polyanskiy, Y.; Poor, H.V.; Verdú, S. Channel coding rate in the finite blocklength regime. *IEEE Trans. Inf.*
643 *Theory* May **2010**, *56*, 2307–2359.