Algebraic Security Analysis of Key Generation with Physical Unclonable Functions

Matthias Hiller[1], Michael Pehl[1], Gerhard Kramer[2] and Georg Sigl[1,3]

[1] Chair of Security in Information Technology
[2] Chair of Communications Engineering
Technical University of Munich
[3] Fraunhofer AISEC
{matthias.hiller, m.pehl, gerhard.kramer, sigl}@tum.de

Abstract. Physical Unclonable Functions (PUFs) provide cryptographic keys for embedded systems without secure non-volatile key storage. Several error correction schemes for key generation with PUFs were introduced, analyzed and implemented over the last years. This work abstracts from the typical algorithmic level and provides an algebraic view to reveal fundamental similarities and differences in the security of these error correction schemes.
An *algebraic core* is introduced for key generation with Physical Unclonable Functions (PUFs). It computes the secret key through the helper data from the input PUF response and an optional random number. For nearly uniformly distributed PUF responses, the leakage of the secret key and the helper data can be brought to zero if and only if the rank of the algebraic core is equal to the sum of the ranks of the key generating part and the rank of the helper data generating part. This rank criterion has the practical advantage that a security check can be performed for linear codes at an early design stage of an algorithm. The criterion is applied to state-of-the-art approaches to show that fuzzy commitment and systematic low leakage coding are the only analyzed schemes that achieve zero leakage.

Keywords: Physical Unclonable Functions (PUFs), Fuzzy Extractor, Coding Theory.

1 Introduction

Physical circuit properties such as the exact individual switching delays of transistors vary for each manufactured chip. While conventional circuits suffer from this variation, silicon *Physical Unclonable Functions (PUFs)* take advantage of them: they capture randomness in the manufacturing process and transform the analog physical variations into digital numbers which can be interpreted as outcome of a random variable. These results can be used to embed a key into a device and only reproduce it on demand [1].

Since silicon PUFs are constructed from transistors, or even from standard cells, their implementation fits in seamlessly with the standard digital design flow and manufacturing process. Therefore, PUFs can be easily added to a standard integrated circuit and bridge the gap between the increasing demand for security and the restriction of a low additional cost overhead.

In this work, we focus on the key storage application of PUFs where the PUF provides a sequence of bits, called the *PUF response*, which is derived by sampling and quantizing the analog circuit behavior. The circuit can be implemented e.g. through ring oscillators [2], SRAM cells [3], or configurable delay lines with an arbiter [4]. Once the PUF is manufactured and the variations are fixed to a specific value, a noisy version of the initial PUF response can be reproduced whenever needed.

To overcome the noise in the PUF response and generate sufficiently reliable keys, with error probabilities in the range of 10^{-6} down to 10^{-9}, error correction is required. The building blocks in Figure 1 show the processing steps for secret key storage with PUFs during enrollment which is carried out in a secure environment. Later, the secret key is reproduced again in the field.

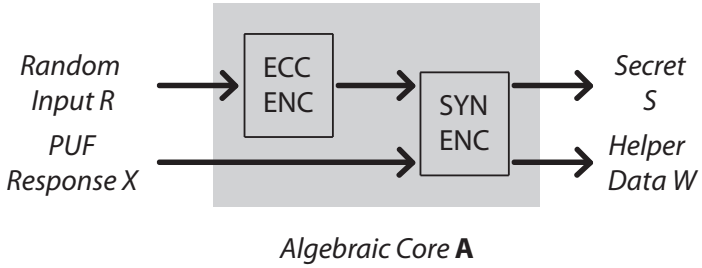

Fig. 1: Secret Key and Helper Data Generation with a PUF during enrollment

Error correction is enabled by mapping the initial PUF response to a codeword of an *Error-Correcting Code* (ECC) during enrollment. First, a random number is encoded to a codeword. Then, the syndrome encoder creates the secret and side-information, called *helper data*, from the PUF response and the codeword. The helper data is stored for error correction later during reproduction. Later in this work, we will merge the ECC and syndrome encoder and treat both together as *algebraic core* highlighted in gray in Figure 1.

In this work, we are interested in the secrecy leakage between the secret and the helper data and, thus, assume a good linear ECC as given. Over the last years, several practical algorithms were introduced and implemented, e.g. [5,6,7,8,9,10,11]. So far, the development of new error correction algorithms for PUFs was driven more from a design perspective and new solutions were compared to the state of the art by analyzing practical quantities such as error probabilities, leakages and implementation sizes. The information theory community studied the more generic, but closely related, problem of key generation

from a correlated source [12,13,14], which led to capacity results and optimal coding strategies based on random codes.

We introduce the algebraic core as a unified algebraic representation of the process of secret and helper data generation which applies to most currently available schemes. Our new unified description allows to discover similarities and differences among most currently available schemes which is in line with previous work [10,15]. Instead of aiming for a more precise analysis of the leakage, we provide rather qualitative, but more constructive results. We provide a new view on the problem which allows to identify the reason for the leakage already during the design of the algorithm.

The ranks of different submatrices of the algebraic core allow to analyze properties of state-of-the-art approaches and are a starting point for a systematic exploration to develop better solutions.

Analyzing the state-of-the-art approaches reveals that only the cores of fuzzy commitment [6] and systematic low leakage coding [16,10] fulfill our security criterion while the other approaches have inevitable secrecy leakage that is already caused on algorithmic level, which is in line with e.g. [10,15].

Our Contributions:

- Generic algebraic representation of syndrome and channel coding for PUFs
- Discussion of the algebraic cores of seven coding schemes
- Introduction of a practical security design criterion
- Algebraic security analyses of state-of-the-art approaches

Outline:
The notation used in this paper is provided in Section 2. Section 3 introduces our new algebraic representation of helper data generation for PUFs. The algebraic representations of several state-of-the-art schemes are given in Section 4. We derive our security criterion in Section 5 and apply it in Section 6. Section 7 concludes this work.

2 Notation

Functions are denoted by small letters, e.g., $f(\cdot)$. Line vectors are denoted by capital letters, e.g., X, where $[X, Y]$ denotes the concatenation of vectors X and Y to a new line vector. Matrices are provided in bold capital letters, e.g., $\mathbf{A}$. The dimension of a vector or matrix is denoted by $dim(\cdot)$ and evaluates, e.g., to $\dim(\mathbf{A}) = \langle a \times b \rangle$, where the first entry (a) is the number of rows and the second entry (b) is the number of columns. The rank of a matrix is computed by the $rank\,(\cdot)$ operation, a diagonal matrix is created from a vector X by applying $diag\,(X)$.

The entropy can be evaluated to $H(X)$, the joint entropy is given by $H(XY)$. $I(X;Y)$ denotes the mutual information between.

$\mathbf{I}$ is the identity matrix. $\mathbf{G}$ defines the generator matrix of a code with the special case $\mathbf{G} = [\mathbf{I}\,\mathbf{P}]$ for a linear code in systematic form. $\mathbf{P}$ is the part of

the generator matrix of a linear code in standard form which creates to the redundancy while $\mathbf{H}$ is the parity check matrix of a code.

3 Unified Algebraic Description of Secret Key and Helper Data Generation

Currently, most helper data generation schemes are presented in an algorithmic way. This representation is closer to implementation but makes it harder to compare their fundamental theoretical properties. Therefore, we introduce a unified algebraic form which gives better insights on the underlying structure, why and how much the helper data leaks for different coding schemes.

The PUF response X obviously is a mandatory input for all secret key storage approaches for PUFs. Some approaches also have a random number R as second input that is available only during helper data generation. It is either directly used as the key or as a mask to protect the PUF response.

The helper data generation has two outputs. The secret S is either directly used as key K or compressed by a hash function $f(\cdot)$. In addition, helper data W is stored to enable later key reproduction.

Helper data generation schemes for PUFs with an algebraic core $\mathbf{A}$ share the form

$$[S\ W] = [R\ X]\ \mathbf{M}_{pre}\ \mathbf{A}\ \mathbf{M}_{post} \tag{1}$$

In the first step, the random number R and PUF response X with dimensions $\langle 1 \times k_{in} \rangle$ and $\langle 1 \times l_{in} \rangle$ are multiplied with a preprocessing matrix $\mathbf{M}_{pre}$. The algebraic core $\mathbf{A}$ performs the fundamental encoding operations. The result is multiplied with the postprocessing matrix $\mathbf{M}_{post}$. The outputs S and W have dimensions $\langle 1 \times k_{out} \rangle$ and $\langle 1 \times l_{out} \rangle$.

Some schemes read in reliability information on the PUF response bits and select the most stable ones [17] during preprocessing, or apply postprocessing steps that take the reliability into account, such as [8,9]. These algorithmic steps can be represented in the preprocessing matrix $\mathbf{M}_{pre}$ or postprocessing matrix $\mathbf{M}_{post}$. We require that there is no interaction between R and X in $\mathbf{M}_{pre}$ and also no interaction between the preliminary versions of S and W in $\mathbf{M}_{post}$. If there is interaction, we have to consider the processing step as part of the algebraic core.

Some approaches can be attacked by manipulating the helper data [18,19]. A hash function $f(\cdot)$ can prevent these attacks by combining the helper data and the secret via $K = f(S, W)$ [18]. The helper data leaks information about the reproduced secret for some approaches. If this is the case, a hash function is required to compress the remaining entropy $H(S|W)$ to a smaller length to create a high-quality cryptographic key $K = f(S)$ [7]. According to [7], information theoretic security can be achieved by using a universal hash function. However, the hash loss [20] must be taken into account which has to be compensated by a larger secret S. Ideally, we try to avoid using a hash function so that no hash loss occurs and also to reduce the implementation complexity.

In this work, we focus on the properties of the algebraic core and remove $\mathbf{M}_{pre}$ and $\mathbf{M}_{post}$, which can be set to $\mathbf{I}$ for reliable PUFs. Therefore, (1) simplifies to

$$[S\ W] = [R\ X]\ \mathbf{A} \tag{2}$$

The algebraic core $\mathbf{A}$ has a left part $\mathbf{A}_L$ that computes the secret S and a right part $\mathbf{A}_R$ that computes the helper data W. The four sub-matrices of interest are given by

$$\mathbf{A} = [\mathbf{A}_L\ \mathbf{A}_R] = \begin{bmatrix} \mathbf{A}_{UL}\ \mathbf{A}_{UR} \\ \mathbf{A}_{LL}\ \mathbf{A}_{LR} \end{bmatrix} \tag{3}$$

Inserting (3) in (2) shows that R is multiplied with the upper part $[\mathbf{A}_{UL}\ \mathbf{A}_{UR}]$ of $\mathbf{A}$ while input X is multiplied with the lower part $[\mathbf{A}_{LL}\ \mathbf{A}_{LR}]$.

4 Algebraic Representation of the State of the Art

This section describes the algebraic cores for five linear key generation approaches with PUFs discussed in [15] as a reference set. The analysis of the leakage of the different approaches follows in Section 5.

4.1 Fuzzy Commitment

For fuzzy commitment [6] using a linear ECC, a random input vector R is stored with the help of the PUF and is used as the secret S. By multiplying R with the generator matrix $\mathbf{G}$ of an ECC, R is encoded to a codeword $S = C = R\ \mathbf{G}$. Then, it is masked with the PUF response X and the result is stored as public helper data W, i.e., $W = (R\ \mathbf{G}) \oplus X$. Note that fuzzy commitment and the code-offset fuzzy extractor can also operate on non-linear ECCs. However, all published implementations in the PUF context use linear ECCs.

Let decode$(\cdot)$ be the decoding operation of the ECC. The resulting equations in matrix form are

$$[S\ W] = [R\ X]\ \begin{pmatrix} \mathbf{G}\ \mathbf{G} \\ \mathbf{0}\ \mathbf{I} \end{pmatrix} \tag{4}$$

$$K = \text{decode}(S) = R \tag{5}$$

There also exists a slightly modified version with $K = S = R$, presented in [21]. Here, $\mathbf{A}_{UL} = \mathbf{G}$ is replaced by $\mathbf{I}$, resulting in

$$[S\ W] = [R\ X]\ \begin{pmatrix} \mathbf{I}\ \mathbf{G} \\ \mathbf{0}\ \mathbf{I} \end{pmatrix} \tag{6}$$

Note that [6] returns the entire codeword that also contains redundancy while [21] only outputs the encoded information.

4.2 Code-Offset Fuzzy Extractor

The code-offset fuzzy extractor [7] has several parallels to fuzzy commitment. The main difference is that the PUF response X defines the secret S instead of deriving it from the random number R. This difference causes leakage so that a hash function must be added. For a code-offset fuzzy extractor using a linear ECC, the helper data W is computed as the XOR of a random codeword $R\ \mathbf{G}$ and the PUF response X. Since the PUF response is used as the secret now, i.e. $S = X$, we have

$$[S\ W] = [R\ X] \begin{pmatrix} \mathbf{0} & \mathbf{G} \\ \mathbf{I} & \mathbf{I} \end{pmatrix} \tag{7}$$

$$K = f(S) \tag{8}$$

4.3 Fuzzy Extractor with Syndrome Construction

A second construction introduced in [7] stores the syndrome of the PUF response as helper data so that no extra input random number R is required. Note that for ECCs, the syndrome is precisely defined, e.g. for block codes with parity check matrix $\mathbf{H}$ it is given by $\mathbf{H}\ X^T$. In the general PUF context, we interpret the word syndrome as information that facilitates error correction since it also contains information on the error pattern in the PUF response. Here, we use *syndrome construction* to refer to the channel coding definition.

We compute the syndrome by multiplying the PUF response X with $\mathbf{H}^T$. Again, a hash function $K = f(S)$ compresses the PUF output to create a uniform cryptographic key.

Since no random number R is used, the two upper sub-matrices of $\mathbf{A}$ are set to zero. It can be seen that all PUF response bits contribute to the helper data and also to the key. The unified algebraic representation of the syndrome construction is given by

$$[S\ W] = [0\ X] \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{I} & \mathbf{H}^T \end{pmatrix} \tag{9}$$

$$K = f(S) \tag{10}$$

4.4 Parity Construction

Instead of storing the syndrome, the construction in [5] stores the parity of the PUF response. The entire PUF response is interpreted as information to be encoded by an ECC with systematic encoding with $\mathbf{G} = (\mathbf{I}\ \mathbf{P})$, including the parity part $\mathbf{P}$. As for the syndrome construction, the secret and the helper data are computed from the PUF response while no external secret is used. Therefore, $\mathbf{A}_{UL}$ and $\mathbf{A}_{UR}$ are both set to zero again. The unified mathematical description is

$$[S\ W] = [0\ X] \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{I} & \mathbf{P} \end{pmatrix} \tag{11}$$

$$K = f(S) \tag{12}$$

It was shown in [15] that the error correction capability of the parity construction is significantly weaker than the other discussed approaches.

4.5 Systematic Low Leakage Coding

Systematic low leakage coding (SLLC) [16,10] is the newest approach in this comparison and is also based on codes with systematic encoding. SLLC also computes redundancy from PUF response bits.

Similar to the two previous approaches, SLLC has an algebraic core in the form

$$[S\ W] = [\mathbf{0}\ X] \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{LL} & \mathbf{A}_{LR} \end{pmatrix} \tag{13}$$

In contrast to the parity approach, the parity is not stored directly. Instead, it is masked with fresh PUF bits and then output as helper data. Therefore, we split the PUF response X into a first part $K = S = X_1^{k_{out}}$ that is used as secret and a second part $X_{k_{out}+1}^{l_{in}}$ for masking. $\mathbf{A}_{LL}$ and $\mathbf{A}_{LR}$ in (13) are given by

$$\mathbf{A}_{LL} = \begin{pmatrix} \mathbf{I} \\ \mathbf{0} \end{pmatrix}, \quad \mathbf{A}_{LR} = \begin{pmatrix} \mathbf{P} \\ \mathbf{I} \end{pmatrix} \tag{14}$$

Inserting (14) in (13) and removing the all zero row gives

$$[S\ W] = \left[X_1^{k_{out}}\ X_{k_{out}+1}^{l_{in}}\right] \begin{pmatrix} \mathbf{I} & \mathbf{P} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \tag{15}$$

Note the similarity to (6), where $\mathbf{A}$ is also an upper triangular matrix.

5 Security Criterion for the Algebraic Core

We next discuss the secrecy leakage that is quantified by the mutual information $I(S;W)$. Our goal is to make this leakage close to zero, in which case S and W are independent. We show that this can be achieved only if the rank of the algebraic core $\mathbf{A}$ is equal to the sum of the ranks of $\mathbf{A}_L$ and $\mathbf{A}_R$. For the analysis, we will define the rank loss as the difference between the maximum possible rank of a matrix and its actual rank.

For two sequences $Y \in \mathbb{F}_2^n$ and $Z \in \mathbb{F}_2^m$ and a matrix $\mathbf{A}$ with $dim(\mathbf{A}) = \langle n \times m \rangle$ and $q = rank(\mathbf{A})$, we compute Z as $Z = Y \cdot \mathbf{A}$. This can be interpreted as mapping $Y \in \mathbb{F}_2^n$ into a subspace $\mathcal{Q} \subseteq \mathbb{F}_2^m$ with $|\mathcal{Q}| = 2^q$ points. $H(Z) = q$ if all points in $\mathcal{Q}$ occur with the same probability and $H(Z) < q$, otherwise. The entropy of Z is at most the rank of $\mathbf{A}$.

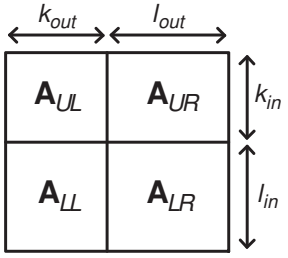

Fig. 2: Dimensions of the Sub-matrices of the Algebraic Core

Let $\mathcal{R}$ be the set of all input random vectors R and $\mathcal{X}$ be the set of all PUF responses X. Further let $\mathcal{S}$ contain all secrets S and $\mathcal{W}$ all helper data W. The algebraic core $\mathbf{A}$ maps elements of the input set $\mathcal{R} \times \mathcal{X} = \mathbb{F}_2^{k_{in}+l_{in}}$ to elements of the output spaces $\mathcal{S} \times \mathcal{W} \subseteq \mathbb{F}_2^{k_{out}+l_{out}}$ where $\log_2 |\mathcal{S} \times \mathcal{W}|$ is at most the rank of $\mathbf{A}$. If elements in $\mathcal{R} \times \mathcal{X}$ occur with probability zero, then $\log_2 |\mathcal{S} \times \mathcal{W}| < rank\,(\mathbf{A})$. Note that the rank is the maximum number of base vectors spanning the output space. Figure 2 shows the dimensions of the sub-matrices of $\mathbf{A}$.

Spaces $\mathcal{S}$ and $\mathcal{W}$ have bases formed by rows $A_{L,i}$ and $A_{R,i}$ of $\mathbf{A}_L$ and $\mathbf{A}_R$, respectively. By definition, the vectors in each index set $\mathcal{I}$ are linearly independent such that any linear combination of base vectors can be zero only if all coefficients λ_i are zero. The corresponding index sets contain the indices of the base vectors such that

$$\mathcal{I}_L = \left\{ i \in \{1, ..., k_{in} + l_{in}\} \,\middle|\, \sum_i \lambda_i A_{L,i} = 0 \Leftrightarrow \forall i : \lambda_i = 0 \right\}$$
$$\mathcal{I}_R = \left\{ i \in \{1, ..., k_{in} + l_{in}\} \,\middle|\, \sum_i \lambda_i A_{R,i} = 0 \Leftrightarrow \forall i : \lambda_i = 0 \right\}$$

with $|\mathcal{I}_L| \leq rank\,(\mathbf{A}_L)$ and $|\mathcal{I}_R| \leq rank\,(\mathbf{A}_R)$. Accordingly, we get for the full algebraic core $\mathbf{A}$

$$\mathcal{I} = \left\{ i \in \{1, ..., k_{in} + l_{in}\} \,\middle|\, \sum_i \lambda_i A_i = 0 \Leftrightarrow \forall i : \lambda_i = 0 \right\}$$

with $|\mathcal{I}| \leq rank\,(\mathbf{A})$.
In the following, we will use the difference between the maximum rank of a matrix and its actual rank to analyze the relation between the overall core $\mathbf{A}$ and its components $\mathbf{A}_L$ and $\mathbf{A}_R$. We define the rank losses as

$$\Delta_L = \min\{k_{in} + l_{in}, k_{out}\} - rank\,(\mathbf{A}_L) \tag{16}$$
$$\Delta_R = \min\{k_{in} + l_{in}, l_{out}\} - rank\,(\mathbf{A}_R) \tag{17}$$
$$\Delta = \min\{k_{in} + l_{in}, k_{out} + l_{out}\} - rank\,(\mathbf{A}) \tag{18}$$

In addition, we define the minimum rank loss $g_{min}\,(k_{in} + l_{in}, k_{out}, l_{out})$ of $\mathbf{A}$ as

$$\begin{aligned} g_{min}\,(k_{in} + l_{in}, k_{out}, l_{out}) = &\min\{k_{in} + l_{in}, k_{out}\} \\ &+ \min\{k_{in} + l_{in}, l_{out}\} \\ &- \min\{k_{in} + l_{in}, k_{out} + l_{out}\} \end{aligned} \tag{19}$$

Only if there are rows of $\mathbf{A}_L$ and $\mathbf{A}_R$ that form bases of two complementary vector spaces with dimensions $rank\,(\mathbf{A}_L)$ and $rank\,(\mathbf{A}_R)$, i.e., the indices of the rows selected for the bases from $\mathbf{A}_L$ and $\mathbf{A}_R$ are different, the secrecy leakage can be brought to zero. Only then, S and W can become independent. The two vector spaces together form the vector space built by a basis formed by rows of $\mathbf{A}$. The difference between the minimum rank loss and the actual rank loss, i.e.,

$$\begin{aligned} &g_{min}\,(k_{in} + l_{in}, k_{out}, l_{out}) \;-\; (\Delta_L + \Delta_R - \Delta) \\ &= rank\,(\mathbf{A}_L) + rank\,(\mathbf{A}_R) - rank\,(\mathbf{A}) \end{aligned} \tag{20}$$

determines the mutual information $I\,(S; W)$ between the secret and helper data. Our goal is to bound this mutual information by some small ϵ_0 which depends on the entropy of the input PUF data $H(X)$ and the entropy of the random number $H(R)$.

For this purpose, we select from all possible index sets all those which describe a maximum set of linearly independent rows. This is fulfilled if index sets are selected, for which

$$|\mathcal{I}_L| + |\mathcal{I}_R| = rank\,(\mathbf{A}_L) + rank\,(\mathbf{A}_R) \tag{21}$$

The selected index sets can be used to construct index sets for $\mathbf{A}$.

$$\mathcal{I} = \mathcal{I}_L \cup \mathcal{I}_R \tag{22}$$

All these sets are searched for a set $\mathcal{I}$ which is built from non-overlapping sets $\mathcal{I}_L$,$\mathcal{I}_R$. This is ensured iff (21), (22) and

$$|\mathcal{I}| = |\mathcal{I}_L| + |\mathcal{I}_R| \tag{23}$$

hold. If such a set $\mathcal{I}$ exists,i.e.,

$$\begin{aligned} &\underset{\mathcal{I},\mathcal{I}_L,\mathcal{I}_R}{\exists} (\mathcal{I} = \mathcal{I}_L \cup \mathcal{I}_R) \\ &\wedge (|\mathcal{I}| = |\mathcal{I}_L| + |\mathcal{I}_R| = rank\,(\mathbf{A}_L) + rank\,(\mathbf{A}_R)) \end{aligned} \tag{24}$$

it is ensured, that no information leaks due to the structure of the syndrome coding approach. However, if (24) cannot be fulfilled, information is leaked. Thus, we claim that (24) is a necessary and sufficient condition to ensure $I(S;W) \leq \epsilon_0$. As soon as there is any overlap in the index sets, S and W cannot be independent. The difference $\Delta_L + \Delta_R - \Delta$ is increased by one for each overlapping index so that the leakage of the algebraic core is increased by one accordingly. The mutual information $I(S;W)$ is the quantity of interest and is given by

$$I(S;W) = H(S) + H(W) - H(S\ W) \tag{25}$$

Based on the observations regarding the rank loss, we compute the mutual information of secret and helper data using the entropy of the inputs and the rank of the algebraic core. Let $\epsilon, \epsilon_L, \epsilon_R > 0$, sufficiently large. Then we can rewrite the components of (25) as

$$\begin{aligned} H(S) =& H([R\ X]\ \mathbf{A}_L) \\ =& rank(\mathbf{A}_L) - \epsilon_L \\ =& \min\{k_{in} + l_{in}, k_{out}\} - \Delta_L - \epsilon_L \end{aligned} \tag{26}$$

$$\begin{aligned} H(W) =& H([R\ X]\ \mathbf{A}_R) \\ =& rank(\mathbf{A}_R) - \epsilon_R \\ =& \min\{k_{in} + l_{in}, l_{out}\} - \Delta_R - \epsilon_R \end{aligned} \tag{27}$$

$$\begin{aligned} H(S\ W) =& H([R\ X]\ \mathbf{A}) \\ =& rank(\mathbf{A}) - \epsilon \\ =& \min\{k_{in} + l_{in}, k_{out} + l_{out}\} - \Delta - \epsilon \end{aligned} \tag{28}$$

We define the overall loss ϵ_0 as $\epsilon_0 = \epsilon - \epsilon_L - \epsilon_R$. Sources with high entropy have $H(R) = k_{in} - \epsilon_{in,1}$ and $H(X) = l_{in} - \epsilon_{in,2}$ with small $\epsilon_{in,1}, \epsilon_{in,2} > 0$. As worst case assumption, we can take $\epsilon = \epsilon_{in,1} + \epsilon_{in,2}$ and $\epsilon_L = \epsilon_R = 0$ such that $\epsilon_0 = \epsilon_{in,1} + \epsilon_{in,2}$. This value depends only on the PUF implementation so that it is the responsibility of the PUF designer to provide high-quality PUF data and bring ϵ_0 down. In practice, ϵ_0 values close to zero are achievable for PUF implementations with responses which are distributed closely to uniform [22].

With the given entropies and (25), the secrecy leakage given by the mutual information of S and W is computed as

$$I(S;W) = H(S) + H(W) - H(S\ W) \tag{29}$$

$$= rank(\mathbf{A}_R) + rank(\mathbf{A}_L) - rank(\mathbf{A}) + \epsilon_0 \tag{30}$$

$$= g_{min}(k_{in} + l_{in}, k_{out}, l_{out}) - (\Delta_L + \Delta_R - \Delta) + \epsilon_0 \tag{31}$$

Observe that the mutual information in (30) can only be close to zero if (24) holds.

The state-of-the-art approaches output either R or X as S. Therefore, either $\mathbf{A}_{UL} = \mathbf{I}$ or $\mathbf{A}_{LL} = \mathbf{I}$, while the second sub-matrix of $\mathbf{A}_L$ is set to zero. Afterwards, S is either directly output as key or fed into a hash function. The hash function is necessary for compression if $k_{out} > H(S) + \delta$ for some small $\delta > 0$, or to mitigate leakage if secret information is leaked through the helper data as discussed in (31).

6 Security Analysis of the State of the Art

In this section, we apply our rank criterion to the algebraic cores discussed in Section 4.

6.1 Fuzzy Commitment

The core of fuzzy commitment in (6) is an upper triangular matrix with full rank such that (24) is fulfilled by design. Therefore, the secrecy leakage of the algorithm depends only on the joint entropy of the PUF response X and of the random number R. If it is sufficiently high, the secret S can directly be used as a key K. Otherwise S can be compressed to a smaller key K with a hash function to achieve a sufficiently high entropy per bit in K. The helper data has a fixed size l_{in} which is larger than necessary as discussed in [10].

6.2 Code-Offset Fuzzy Extractor

The left and right parts of the core $\mathbf{A}$ in (7) both have full rank such that $\Delta_L = 0$ and $\Delta_R = 0$ according to (26). However, their index sets overlap. Assigning the lower l_{in} rows to the left side leaves only k_{in} indices for the right side. As a consequence, $\Delta = l_{in} - k_{in}$ and up to $l_{in} - k_{in}$ bits leak, which is consistent with the findings in [23]. I.e., if an attacker knows the helper data, the entropy of an l_{in}-bit long secret is reduced to k_{in}. As a consequence the hash function in (7) must be designed so that the remaining k_{in} bits of entropy are distributed equally to the bits of a k_{in} bit long key.

Note that for fuzzy commitment, the codeword is masked with the PUF response such that it forms a secure one time pad. For the fuzzy extractor the PUF response is masked with the codeword resulting in an imperfect one time pad because by definition, not all bits in the codeword are independent. This small design difference leads to a different secrecy leakage.

6.3 Fuzzy Extractor with Syndrome Construction

For PUF size l_{in} and a code size of k_{out}, this approach uses only $l_{in} - k_{out}$ bits of helper data which is the lowest possible number for a given ECC and thus the best possible solution. In (9), we have $rank\,(\mathbf{A}_{LL}) = l_{in}$ and, accordingly, $rank\,(\mathbf{A}_{UR}) = l_{in} - k_{out}$. However, the index sets overlap fully such that $\Delta = l_{in} - k_{out}$. Therefore, the maximum leakage is the same as for the code-offset fuzzy extractor and again a hash function is required.

Key generation Scheme	Δ	$I(S;W)$ (nearly perfect PUF)
Fuzzy Commitm.	0	$< \epsilon_0$
Code-Offset	$l_{in} - k_{out}$	$< l_{in} - k_{out} + \epsilon_0$
Syndrome Constr.	$l_{in} - k_{out}$	$l_{in} - k_{out}$
Parity Constr.	$l_{in} - k_{out}$	$2k_{out} - l_{in}$
SLLC	0	$< \epsilon_0$

Table 1: Mutual information between S and W of the state-of-the-art syndrome coding approaches for PUFs

6.4 Parity Construction

In (11), the rank of $\mathbf{I}$ is equal to the length of the secret, so $rank\,(\mathbf{I}) = k_{out}$. $\mathbf{P}$ has rank $\min\{k_{out}, l_{in} - k_{out}\}$. As for the previous scheme, the index sets fully overlap such that $rank\,(\mathbf{A}) = k_{out}$. The mutual information is given by $I\,(S;W) = \min\{k_{out}, l_{in} - k_{out}\} + \epsilon_0$ and again, $l_{in} - k_{out}$ bits leak such that a hash function is required.

Note that in (11) only $\max\{0, k_{out} - (l_{in} - k_{out})\}$ secret bits remain so that this approach is only suitable for ECCs with small redundancies $l_{in} - k_{out}$ such that $2k_{out} - l_{in} > 0$ still holds and any secret can be extracted.

6.5 Systematic Low Leakage Coding

Similar to the fuzzy commitment, $\mathbf{A}$ in (15) is an upper triangular matrix. The left and right part of the algebraic core as well as the concatenation of both all have full rank which gives $\Delta = 0$. As a result, the mutual information $I\,(S;W) < \epsilon_0$, i.e., no information leaks due to the structure of the algebraic core. Therefore, SLLC combines the advantage of zero secrecy leakage with the advantage of a minimal helper data size of the syndrome construction.

6.6 Summary on State-of-the-Art Syndrome Decoders

We have shown that most state-of-the-art helper data generation schemes can be brought into a unified algebraic form which allows a comparison of the individual properties.

Wrapping up the results of this section, Table 1 provides an overview over the discussed approaches. Only fuzzy commitment and SLLC have a rank loss Δ of zero while all other approaches have $\Delta = l_{in} - k_{out}$.

We also discuss leakages for a nearly perfect PUF with $H(X) = l_{in} + \epsilon_0$ and perfect random number R with $H(R) = k_{out}$ that show the optimal case. Preprocessing can support to achieve such high entropies. The right column shows results according to (31).

The right column on leakage clearly shows that the approaches which have algebraic cores with full rank do not leak significant secret information.

Our rank criterion allows to evaluate solutions at an early design stage and determine whether an algorithm can achieve zero leakage or not. The rank loss difference gives an upper bound for the secrecy leakage and therefore specifies the minimum requirements for a subsequent hash function.

We provide a generic property that allows to analyze new, more complex and potentially more efficient, practical structures with less obvious leakages in future work. Especially, the currently very regular matrix structures with many identity matrices can be extended to other constructions under the constraint of keeping the rank loss close to zero.

7 Conclusions

Several algorithms for secret key generation with PUFs were proposed to enable error correction by storing helper data and using ECCs. We have brought most state-of-the-art approaches into a generic algebraic representation that reveals security properties on a high level of abstraction.

We have shown that the rank of the algebraic core plays a key role for the security of linear schemes. Only if the left and right parts of the core have independent index sets, the secrecy leakage between key and helper data can be brought to zero. Fuzzy commitment and systematic low leakage coding were the only approaches which fulfill that criterion.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful feedback. This work was partly funded by the German Federal Ministry of Education and Research (BMBF) in the project SIBASE through grant number 01IS13020A and the German Research Foundation (DFG) through grant number KR3517/6-1.

References

1. C. Herder, M. Yu, F. Koushanfar, and S. Devadas, "Physical unclonable functions and applications: A tutorial," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1126–1141, 2014.
2. G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *ACM/IEEE Design Automation Conference (DAC)*, 2007, pp. 9–14.
3. J. Guajardo, S. S. Kumar, G. J. Schrijen, and P. Tuyls, "FPGA intrinsic PUFs and their use for IP protection," in *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, ser. LNCS, P. Paillier and I. Verbauwhede, Eds., vol. 4727. Springer Berlin / Heidelberg, 2007, pp. 63–80.
4. B. Gassend, D. Clarke, M. v. Dijk, and S. Devadas, "Delay-based circuit authentication and applications," in *ACM Symposium on Applied Computing (SAC)*, 2003, pp. 294–301.
5. G. I. Davida, Y. Frankel, and B. J. Matt, "On enabling secure applications through off-line biometric identification," in *IEEE Symposium on Security and Privacy (S&P)*, 1998, pp. 148–157.

6. A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in *ACM Conference on Computer and Communications Security (CCS)*, 1999, pp. 28–36.
7. Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Advances in Cryptology (EUROCRYPT)*, ser. LNCS, C. Cachin and J. L. Camenisch, Eds., vol. 3027. Springer Berlin / Heidelberg, 2004, pp. 523–540.
8. M. Yu and S. Devadas, "Secure and robust error correction for physical unclonable functions," *IEEE Design & Test of Computers*, vol. 27, no. 1, pp. 48–65, 2010.
9. M. Hiller, D. Merli, F. Stumpf, and G. Sigl, "Complementary IBS: Application specific error correction for PUFs," in *IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, 2012, pp. 1–6.
10. M. Hiller, M. Yu, and M. Pehl, "Systematic low leakage coding for physical unclonable functions," in *ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, 2015, pp. 155–166.
11. M. Yu, M. Hiller, and S. Devadas, "Maximum likelihood decoding of device-specific multi-bit symbols for reliable key generation," in *IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, 2015, pp. 38–43.
12. R. Ahlswede and I. Csiszar, "Common randomness in information theory and cryptography - part I: Secret sharing," *IEEE Transations on Information Theory*, vol. 39, no. 4, pp. 1121–1132, 1993.
13. U. Maurer, "Secret key agreement by public discussion from common information," *IEEE Transactions on Information Theory*, vol. 39, pp. 733–742, 1993.
14. H. Boche and R. F. Wyrembelski, "Secret key generation using compound sources - optimal key-rates and communication costs," in *International ITG Conference on Systems, Communications and Coding (SCC)*. IEEE, 2013.
15. J. Delvaux, D. Gu, I. Verbauwhede, M. Hiller, and M. Yu, "Efficient fuzzy extraction of PUF-induced secrets: Theory and applications," in *Conference on Cryptographic Hardware and Embedded Systems (CHES)*, ser. LNCS, B. Gierlichs and A. Poschmann, Eds. Springer Berlin / Heidelberg, 2016.
16. H. Kang, Y. Hori, T. Katashita, M. Hagiwara, and K. Iwamura, "Cryptographic key generation from PUF data using efficient fuzzy extractors," in *International Conference on Advanced Communication Technology (ICACT)*. IEEE, 2014, pp. 23–26.
17. F. Armknecht, R. Maes, A.-R. Sadeghi, B. Sunar, and P. Tuyls, "Memory leakage-resilient encryption based on physically unclonable functions," in *Advances in Cryptology (ASIACRYPT)*, ser. LNCS, M. Matsui, Ed., vol. 5912. Springer Berlin / Heidelberg, 2009, pp. 685–702.
18. M. Hiller, M. Weiner, L. Rodrigues Lima, M. Birkner, and G. Sigl, "Breaking through fixed PUF block limitations with differential sequence coding and convolutional codes," in *International Workshop on Trustworthy Embedded Devices (TrustED)*. ACM, 2013, pp. 43–54.
19. J. Delvaux, D. Gu, D. Schellekens, and I. Verbauwhede, "Helper data algorithms for PUF-based key generation: Overview and analysis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 6, pp. 889–902, 2015.
20. B. Barak, Y. Dodis, H. Krawczyk, O. Pereira, K. Pietrzak, F.-X. Standaert, and Y. Yu, "Leftover hash lemma, revisited," in *Advances in Cryptology (CRYPTO)*, ser. LNCS, P. Rogaway, Ed., vol. 6841. Springer Berlin / Heidelberg, 2011, pp. 1–20.

21. P. Tuyls, A. H. M. Akkermans, T. A. M. Kevenaar, G.-J. Schrijen, A. M. Bazen, and R. N. J. Veldhuis, "Practical biometric authentication with template protection," in *Audio- and Video-Based Biometric Person Authentication (AVBPA)*, ser. LNCS, T. Kanade, A. Jain, and N. Ratha, Eds., vol. 3546. Springer Berlin / Heidelberg, 2005, pp. 436–446.
22. S. Katzenbeisser, U. Kocabas, V. Rozic, A.-R. Sadeghi, I. Verbauwhede, and C. Wachsmann, "PUFs: Myth, fact or busted? a security evaluation of physically unclonable functions (PUFs) cast in silicon," in *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, ser. LNCS, E. Prouff and P. Schaumont, Eds., vol. 7428. Springer Berlin / Heidelberg, 2012, pp. 283–301.
23. T. Ignatenko and F. M. J. Willems, "Biometric security from an information-theoretical perspective," *Foundations and Trends in Communications and Information Theory*, vol. 7, no. 2-3, pp. 135–316, 2012.