New Insights to Key Derivation for Tamper-Evident Physical Unclonable Functions

Vincent Immler[1] and Karthik Uppund[1]

Fraunhofer Institute for Applied and Integrated Security (AISEC), Germany
forename.surname@aisec.fraunhofer.de

Abstract. Several publications presented tamper-evident Physical Unclonable Functions (PUFs) for secure storage of cryptographic keys and tamper-detection. Unfortunately, previously published PUF-based key derivation schemes do not sufficiently take into account the specifics of the underlying application, i.e., an attacker that tampers with the physical parameters of the PUF outside of an idealized noise error model. This is a notable extension of existing schemes for PUF key derivation, as they are typically concerned about helper data leakage, i.e., by how much the PUF's entropy is diminished when gaining access to its helper data.
To address the specifics of tamper-evident PUFs, we formalize the aspect of *tamper-sensitivity*, thereby providing a new tool to rate by how much an attacker is allowed to tamper with the PUF. This complements existing criteria such as effective number of secret bits for entropy and failure rate for reliability. As a result, it provides a fair comparison among different schemes and independent of the PUF implementation, as its unit is based on the noise standard deviation of the underlying PUF measurement.
To overcome the limitations of previous schemes, we then propose an Error-Correcting Code (ECC) based on the *Lee* metric, i.e., a distance metric well-suited to describe the distance between q-ary *symbols* as output from an equidistant quantization, i.e., a higher-order alphabet PUF. This novel approach is required, as the underlying symbols' bits are not i.i.d. which hinders applying previous state-of-the-art approaches.
We present the concept for our scheme and demonstrate its feasibility based on an empirical PUF distribution. The benefits of our approach are an increase by over 21% in effective secret bit compared to previous approaches based on equidistant quantization. At the same time, we improve tamper-sensitivity compared to an equiprobable quantization while ensuring similar reliability and entropy. Hence, this work opens up a new direction of how to interpret the PUF output and details a practically relevant scheme outperforming all previous constructions.

Keywords: Tamper-sensitivity, Physical Unclonable Function (PUF), Fuzzy Extractor, Information Theory, Higher-Order Alphabet PUF (HOA PUF), Lee Metric.

1 Introduction

Tamper-evident PUFs typically evaluate a physical token that obstructs physical access to the IC [TS, EFK+, KA], the whole PCB [IOK+18, VNK+, ION+], or generic enclosures [ZHS], e.g., such a token could be a protective coating or a fully-wrapped envelope around a case containing the device. This token comprises physical parameters that are subject to manufacturing variation and thus, can be used as a PUF, i.e., by measuring these variations and applying suitable algorithmic processing, a cryptographic key is generated that subsequently serves as a Key-Encryption-Key (KEK) [Nat]. Once an attacker attempts to gain access to the device, the token is partially destroyed which irreversibly causes a change in its physical parameters, i.e., ideally the PUF is sufficiently altered by tampering such that the key derivation fails. It is therefore of utmost importance

IACR Transactions on Cryptographic Hardware and Embedded Systems ISSN 2569-2925, Vol. 2019, No. 3, pp. 30–65
DOI:10.13154/tches.v2019.i3.30-65

to optimize this algorithmic part to separate the influence from noise from the effects of tampering. Please note that in this scenario, the physical token is merely used to detect tampering without the practical constraints involved when using battery-backed tamper-detection and response mechanisms [OI].

This concept exceeds the scope of other PUFs commonly implemented in silicon, such as the Ring Oscillator (RO) [SD] or SRAM PUF [GKST, HBF], as these PUFs lack the property of tamper-evidence and only provide key storage that is assumed more secure when compared to non-volatile memory, especially when the device is powered-off. In part, this is owed to the fact that these PUFs are just a component in a larger system and therefore their capability to obstruct physical access is severely limited. As a result, they must still be complemented with other countermeasures such as meshes. Several publications have practically proven that these PUFs lacking tamper-evidence can indeed be attacked with moderate resources available in standard testing labs [HNT+, NSHB, HBNS, TDF+, LTBS].

At their core, all PUFs are based on physical measurement data that is processed by suitable algorithms to generate reliable keys of good cryptographic quality. Some PUF hardware primitives are intrinsically limited to a binary-only output since they are memory based, e.g., the SRAM [GKST, HBF] or Flip-Flop PUF [MTV]. For the ease of implementation, others are designed to map quasi-continuous values from the physical domain to a quantized single-bit response by means of a comparator, as it is the case for the RO-PUF and several others. However, this discards large portions of the information provided by the raw PUF response and does not represent a suitable approach for tamper-evident PUFs. It was shown, e.g., for the tamper-evident Coating PUF [TS], that a multi-bit quantization step increases the output entropy and facilitates a first error-*reduction* step prior to an ECC.

Within the context of tamper-evident PUFs, the raw data as result of the measurement is often non-uniformly distributed, e.g., in case of [TS, IOK+18] it follows a normal distribution. This data is then subject to an error-reduction technique based on a multi-bit quantization scheme and further processed by an ECC. As demonstrated as part of this work, comparing the performance of the overall scheme purely based on their number of effective secret bits and failure rate is insufficient and largely neglects the specifics of tampering. To be more precise, while errors based on an idealized noise model are often effectively counteracted, some of these schemes still allow values to occur that would otherwise be considered improbable due to the chosen noise model. However, as the physical tampering is unconstrained, altered values occur independently of the stochastic model assumed for the noise and thus, a stricter stance must be taken to rule out such values, too.

To address this challenge, we formalize this aspect which we call "tamper-sensitivity" (TS). Subsequently, we rate previously suggested schemes using this notion of TS and showcase that even more advanced ECC schemes fail to outperform a scenario purely based on a well-chosen quantization in that regard. Afterwards, we propose our scheme based on the *Lee* metric and a q-ary channel model which exceeds the scope of a large body of PUF ECC work that with an overwhelming majority focused on Hamming distance and the Binary Symmetric Channel (BSC) including but not limited to [JW, DRS, BGS+, YD, Mae, H].

1.1 Contributions

In short, this work presents the following four contributions:

- A new metric by the name tamper-sensitivity that complements previous properties of PUF-based key derivation such as entropy in effective number of secret bits and failure probability for reliability, in particular for tamper-evident PUFs.

- First application of codes with Lee/Manhattan metric in the domain of PUFs, including updated definitions of Uniqueness and Reliability, two well-known PUF performance criteria to assess empirical PUF data.
- A fair comparison of different PUF key derivation schemes based on different quantization schemes, code constructions such as Code-Offset and Fuzzy Commitment, and code metrics, namely Hamming, Levenshtein, and Lee with regard to aforementioned PUF properties.
- Practical design of a new scheme and comparison to state-of-the-art approaches for tamper-evident PUFs, showcasing a gain of over 21% in effective output secret bits for schemes based on equidistant quantization and a drastically improved tamper-sensitivity when compared to schemes based on equiprobable quantization.

1.2 Organization

A brief outline of our paper is as follows. Related work is discussed in Section 2 which is followed by the formalized description of tamper-sensitivity in Section 3. Subsequently, we introduce our own key derivation scheme in Section 4 based on Limited Magnitude Codes (LMC) and the Lee metric. This new scheme is then evaluated in Section 5 and compared against the state of the art. Eventually, we conclude our work in Section 6.

1.3 Notation

Unless specifically noted otherwise, random variables and their distributions are represented by capital letters, whereas numbers and specific realizations of random variables are denoted as small letters. Subscripts refer to indices of vectors, and superscripts show the length of vectors (in either symbols or bit). $\mathcal{C}$ is the ECC and c stands for an n-bit codeword with k information bits and p parity bits.

For the helper data W of the ECC, a quantized PUF response Y^v with either superscript v as the symbol-wise length with alphabet size q or superscript n as length in bit, the mutual information between PUF response and helper data $\mathrm{I}(Y^v;W)$ measures the information leakage. The *min*-entropy definition for $\tilde{\mathrm{H}}_\infty(Y^v|W)$ is given in [DRS]:

$$\mathrm{I}(Y^v;W) = \mathrm{H}(Y^v) - \mathrm{H}(Y^v|W) \leq v \cdot \log_2(q) - \tilde{\mathrm{H}}_\infty(Y^v|W), \tag{1}$$

$$\tilde{\mathrm{H}}_\infty(Y^v|W) = -\log_2\left(\mathop{\mathrm{E}}_{w}\left[\max_{y^v} \Pr_{Y^v|W}[y^v|w]\right]\right). \tag{2}$$

Throughout this paper, we make use of several distance metrics, namely: $\mathrm{d_E}$ for Euclidean distance, $\mathrm{d_{Lev}}$ for Levenshtein distance, $\mathrm{d_{Lee}}$ for Lee distance, $\mathrm{d_{Man}}$ for Manhattan distance, $\mathrm{d_{H|2}}$ for Hamming distance applied to bit strings, and $\mathrm{d_{H|}}_S$ for Hamming distance applied to strings with symbols of a higher-order alphabet.

2 State of the Art

We align our work with other error-reduction and error-correction techniques for PUFs. To do so, we briefly introduce our PUF system model in Section 2.1. Afterwards, in Section 2.2, we discuss previous work on quantization schemes and bit mappings as means of error-reduction prior to an ECC. Subsequently, in Section 2.3 we briefly consider common ECC proposals for PUFs and explain why they are not suited for our setting.

2.1 Model for Tamper-Evident PUFs

Our PUF system model is illustrated in Figure 1 and represents the practical work of, e.g., [TS, IOK+18, ION+] in sufficient detail to discuss their PUF key derivation specifics that are of general nature and relevant for future proposals of tamper-evident PUFs, too. From left to right, it comprises the tamper-evident PUF and illustrates all necessary steps to generate a key. The upper part represents the enrollment of the PUF, i.e., the point in time when the PUF is initialized in a secure environment and helper data is created to enable later error correction. The lower part depicts the reconstruction in the field where the PUF key is extracted again to serve as secret input for cryptographic applications.

Each single PUF value denoted as X is drawn from its corresponding physical PUF *node*. In both [TS] and [IOK+18], the node from which X is drawn is a capacitor C that is subject to manufacturing variation, i.e., X_1 corresponds to a capacitor C_1, X_2 to C_2, and so on. We specifically refer to this as PUF *node* as opposed to bits, to point out that symbols comprised of multiple bit per node are extracted. This underlying element of a PUF is sometimes also called a PUF primitive and this model is not limited to a specific type of node/primitive. X follows a quasi-continuous Probability Distribution Function (PDF) as illustrated in Figure 2 and is the digital representation of the capacitance obtained by a compensated[1] measurement and subsequent conversion by an Analog-to-Digital Converter (ADC). These compensating techniques, such as [TS, BNTM] depend on the specifics of the PUF and are considered outside the scope of this work. Here, we use the term quasi-continuous since in the actual application we do not know the real value (in the sense of continuous) of the PUF nodes and can only practically measure it using a high-resolution measurement circuit. Therefore, X would be typically represented by an *integer* with its number of bits in binary representation equivalent to the number of bits of the ADC. In total, there are v nodes (i.e., v distinct capacitors) in the PUF and all their values combined are termed PUF *device* and written as X^v, i.e., $X^v = \{X_1, X_2, ..., X_v\}$ with $X \in \mathbb{Z}$.

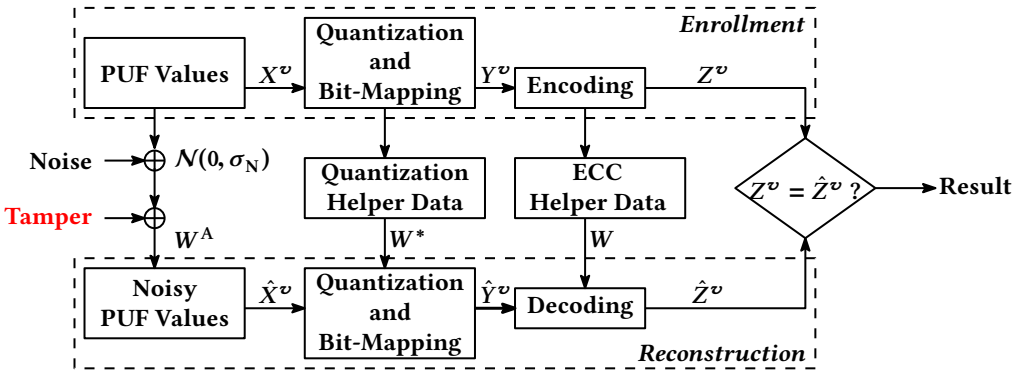


Figure 1: PUF system model with enrollment and reconstruction. Y is the quantized PUF response and Z the secret bit sequence. Added noise is denoted as $\hat{(\cdot)}$.

As part of the data acquisition, the PUF values X are always affected by remaining circuit noise $N \in \mathcal{N}(0, \sigma_N)$ during reconstruction which makes it necessary to account for this influence by suitable mechanisms, e.g., a combination of quantization scheme and ECC. Noise is assumed to be Gaussian following $\mathcal{N}(0, \sigma_N)$, i.e., it is mean free. Moreover, the noise standard deviation σ_N is considered equally distributed across all PUF nodes. If the

[1] The term *compensated* measurement refers to circuit-level techniques to remove temperature and voltage drift effects. An exemplary compensated technique is the 3-signal approach mentioned in [TS]. For other PUF designs, such as the RO-PUF, similar concepts were presented in [BNTM].

system has *not* been tampered with, then the noisy PUF response is $\hat{X}^v = X^v + N^v$. This noise modeling is equivalent to [TS] and also relevant for other systems, such as [ZHS].

Now, in the event of tampering with the PUF, the physical PUF nodes from which values are drawn are additionally altered. This effect is denoted as $W^{\mathrm{A}} \in \mathbb{Z}$, i.e., $\hat{X}^v = X^v + N^v + W^{\mathrm{A}}$ as indicated in Figure 1. We note that W^{A} does not follow a stochastic model or is otherwise formally constrained. This is owed to the fact that a designer of a tamper-evident PUF will not know (*i*) which nodes will be affected by tampering, or how many (*ii*) what the resulting magnitude of the attack is. Hence, regarding the magnitude of W^{A}, we need to implicitly assume that $\sigma_{\mathrm{N}} < W^{\mathrm{A}}$ which is supported by the practical attacks in [TS, IOK+18, ION+]. With respect to the number of nodes affected, it is evident that the best approach will enable tamper detection even in cases when only one node is tampered with. W^{A} is often referred to as shift, and in the noiseless but tampered case, the Euclidean distance $\mathrm{d_E}(X, \hat{X})$ is termed the *tamper magnitude*.

Magnitude of Noise vs. Tampering. Based on this noise model, it is evident that instances in time may occur where $N = 0$ for a specific $\hat{X}$ and at the same time $W^{\mathrm{A}} \approx \sigma_{\mathrm{N}}$, i.e., tampering would go undetected as its magnitude would essentially be mistaken as noise only. Since the noiseless scenario allows for the maximum tamper magnitude to possibly go undetected, this is the scenario we later choose for analysis purposes, i.e., in a real-world implementation, the undetectable tamper magnitude would be smaller as both noise and tampering would occur at the same time. In all other cases, practically speaking, it is similarly difficult to distinguish the noise from the effects of tampering, as an unexpectedly large magnitude may either be the result of a relatively unlikely noise event, or the result of tampering. Hence, the challenge is to devise a scheme that provides a clear Tamper Detection Threshold TDT of whether the error magnitude should be treated as noise, or as tampering, while not impeding typical PUF reliability requirements. This is achieved by schemes where $\mathrm{TDT} = u \cdot \sigma_{\mathrm{N}}$, with u being as small as possible.

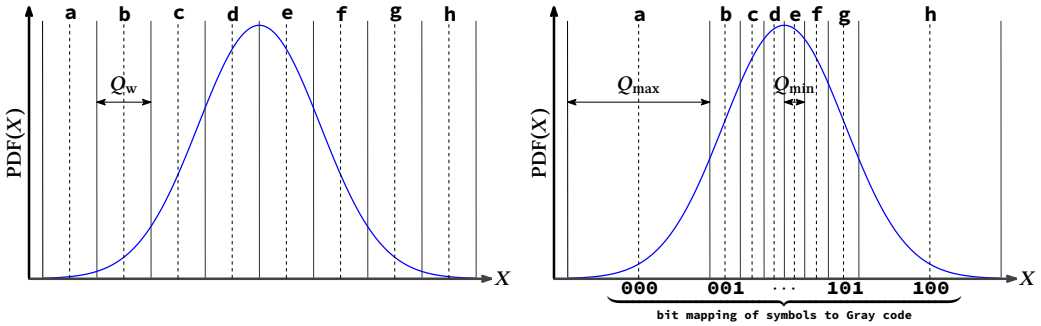
Tamper detection. Here, detection of tampering is the self-determination by the device that $Z^v \neq \hat{Z}^v$ and in that sense no different to the case when the device fails because of insufficient reliability. The interesting result of this work is that ECC schemes effectively working under $\hat{X}^v = X^v + N^v$ are *not* automatically the same to effectively detect the effects of W^{A}, i.e., their TDT is sometimes worse compared to schemes providing much less entropy. As additional constraints, we aim at schemes with superior detection of W^{A} while ensuring the following two requirements:

- The reliability or device failure rate, written as the mismatch probability $\mathrm{P_e}(Z^v) = \Pr[Z^v \neq \hat{Z}^v]$ shall be $< 10^{-6}$ in the presence of noise (without tampering).
- The effective number of secret bits that are extracted from the tamper-evident PUF should be sufficiently large, e.g., $\tilde{\mathrm{H}}_\infty(Y^v|W) > 128$ bits (preferably more).

2.2 Quantization Schemes and Bit Mappings

Thus far, there are two predominant schemes to quantize normally distributed PUF data. Both schemes are based on subdividing the quasi-continuous PDF based on the distribution of X into intervals. In case of equiprobable quantization [TS], the intervals are chosen such that the intervals occur with equal probability. In contrast, equidistant quantization [IHKS] divides it into intervals of equal width. In order to decrease the probability of an erroneous quantization value Y, an offset is stored as helper data W^* during enrollment that shifts the PUF value X to the center of its corresponding quantization interval. For reasons of clarity of explanations, we always assume that symbols of a Higher-Order Alphabet (HOA) are assigned to these intervals as a first processing step even though this was not necessarily included in the original publication, i.e., the PUF output alphabet $\mathcal{L}$ is not $\mathcal{L} = \{0, 1\}$ but $\mathcal{L} = \{a, b, c, d, \ldots\}$, whereas $|\mathcal{L}|$ is the size of the alphabet which is equivalent to

the number of quantization intervals L. Hence, this is referred to as HOA PUF. Both quantization approaches and the assignment of symbols are sketched in Figure 2.



(a) Equidistant quantization. **(b)** Equiprobable quantization.

Figure 2: Visualization of equiprobable and equidistant quantization schemes processing PDF(X) which follows $\mathcal{N}(\mu_X, \sigma_X)$ based on the parameters given in [TS].

Equiprobable quantization of PUF data was first introduced in [TS] for the tamper-evident Coating PUF. As proposed in [TS], each interval is assigned a multi-bit binary representation by means of a Gray code, i.e., neighboring intervals are designed such that their binary representation differs by a one bit substitution error only. Hence, the Hamming distance in binary denoted as $\mathrm{d}_{\mathrm{H}|2}$ is 1 for directly neighboring intervals. Please note that for this approach, both symbols *and* their corresponding binary bit mapping are i.i.d. *and* uniformly distributed. The processed output prior to the ECC is therefore a binary alphabet. However, for the specific scheme presented in [TS], it was later shown that the length of each individual helper data offsets W^* stored for the quantization during enrollment leaks significant amounts of information on the PUF key [IHKS]. In addition, ensuring uniformity of bits requires precise knowledge of the underlying PUF distribution and therefore limits the practical relevance of this scheme.

Other equiprobable quantization schemes implement a partitioning scheme to avoid helper data leakage but again require precise knowledge of the distribution [VTO+, SAS]. Furthermore, as pointed out in [IHKS] and detailed later as part of this paper, equiprobable quantization is ineffective to ensure good tamper-sensitivity in all scenarios due to the size of the outermost intervals of width $Q_{\max}$, as can be deduced from Figure 2b already.

Equidistant quantization apparently mitigates these effects due to the evenly sized intervals with only minor leakage from the sign of its helper data W^*. Moreover, a suboptimal assignment of the interval boundaries relative to the PDF only has an insignificant impact on the resulting entropy of the quantized output, making it a suitable approach, e.g., for [ZHS]. However, it comes at the downside of a biased quantized PUF output, i.e., when mapping the symbols to bits, it is evident that the individual positions of the resulting bit string are neither i.i.d. nor uniform. As a result is any fixed-length binary bit mapping of the symbols heavily biased. Correspondingly, when combining equidistant quantization with a fixed-length binary output and a linear fuzzy extractor scheme, significant amounts of secret information would be leaked by the helper data due to the induced bias [IW, DGV+].

To overcome some of the limitations of equidistant quantization, the authors of [IHL+] proposed a variable-length bit mapping of the symbols. Hence, as a kind of debiasing step, they follow the information theoretic intuition of assigning shorter binary representations to intervals that occur more often, while assigning longer bit representations to intervals that occur less often. However, the quantized sequence comprised of the values Y is no longer of fixed length which necessitates Varshamov-Tenengolts (VT) codes operating in Levenshtein distance $\mathrm{d}_{\mathrm{Lev}}$, accounting not only for substitution errors but also insertions

and deletions [Ten, VT]. This is due to the fact that more commonly known codes such as Bose-Chaudhuri-Hocquenghem (BCH) and Reed Solomon (RS) are not designed to work on variable-length inputs. While VT-codes are well-suited to operate in Levenshtein metric, their overall capability in terms of error-correction is still quite limited. The specific values of each quantization interval are chosen such that neighboring intervals differ by $\mathrm{d}_{\mathrm{Lev}} = 1$ in [IHL+], i.e., the bit mapping of symbols to binary is similar to a Gray code such that directly neighboring intervals differ by only one substitution or insertion/deletion error. Again, a rather precise knowledge and symmetry of the PDF is required to ensure proper behavior of this scheme, otherwise the debiasing properties are degraded.

As later demonstrated as part of our evaluation in Section 5, the scheme based on equidistant quantization and VT-codes also falls short when it comes to tamper-sensitivity when compared to a scenario only based on equidistant quantization without ECC. This is contrary to the claims of [IHL+] and shows that formalizing the aspect of tamper-sensitivity is indeed an important step towards developing more tailored schemes for tamper-evident PUFs. Our scheme presented in Section 4 is based on an equidistant quantization, too. Hence, the subsequent ECC operates on the quantized PUF output Y^v which is based on symbols with aforementioned properties. Please note that the overall setting in this work deviates quite significantly from scenarios commonly assumed, e.g., for the SRAM PUF.

2.3 Error-Correcting Codes for PUFs

A significant amount of work was carried out in the domain of PUFs ranging from formalizing PUFs [AMS+] to generic ECC constructions, and protocols [CBFH] in addition to analyses in terms of implementation and information efficiency [Mae, HYP, DGV+]. As indicated beforehand, previous work is mostly specifically tailored towards PUFs based on a binary alphabet with only very few exceptions [IHKS, IHL+]. The strong focus on these binary-only PUFs has been a valid requirement due to their ease of physical construction in silicon and widespread availability. While generally being suitable to provide a sufficient reliability even for other scenarios than their intended purpose, the shortcoming of most ECC schemes is related to helper data leakage in W that is caused by biased PUF data and/or insufficiencies of the ECC construction, as detailed in [IW, HPKS, DGV+]. If not considered at all, helper data leakage is a severe security threat, as the anticipated security level is not present in the design. If not systematically counteracted on an algorithmic level, helper data leakage impacts the cost/size of the PUF implementation, as demonstrated for example in [H], where – depending on the chosen ECC construction – the corresponding PUF size would differ by a factor of ~ 2 to achieve the same security level. Hence, the problem of bias in PUF data and ECC helper data leakage is not completely new and the same is true for ideas of counteracting it. Therefore, when considering new ECC approaches for tamper-evident PUFs and higher-order alphabets, these known effects and existing concepts must be taken sufficiently into account as done in the following.

To remove PUF induced leakage, various debiasing schemes were proposed. Index-Based Syndrome coding (IBS) [YD] is a debiasing technique that also improves the reliability by indexing only reliable PUF response bits. However, the symbols of an equidistant quantization as later used in our scheme all have the same reliability such that IBS is not applicable to the discussed scenario. Moreover, not considering certain bits of the PUF output counteracts the idea of detecting tamper attempts.

The scheme presented in [MvdLvdSW] improves the von Neumann (VN) corrector [vN]. For i.i.d. PUF response bits (which is different to our scenario), pairs of consecutive zeros or ones occur with different probabilities, while pairs (1,0) and (0,1) have the same probability. However, the approach is intended for PUFs with small output alphabets. It evaluates groups of elements that occur with the same probability but differ in their sequence, such that an increasing number of elements decreases the probability of these equiprobable events. In [SUHA], it was extended to ternary outputs using reliability information. However, it cannot be efficiently applied to higher-order alphabets. The

multi-bit symbol approach in [YHD] is especially suited for PUFs with high bit error probabilities $> 20\%$. It is not explicitly designed for bias reduction but can also handle biased inputs efficiently as well. Additional recent debiasing work includes [H] where again the PUF bits are assumed i.i.d. and coset coding is applied to mitigate the leakage. This idea could be interpreted as combining different equidistant quantization intervals to create a more uniform occurrence of the symbols. However, this again would contradict the idea of tamper-sensitivity as will become evident by the remainder of the paper.

As a result, none of the discussed techniques specifically address biased symbols of a higher-order alphabet. Hence, while they may work on such data, they are designated to perform less efficiently when deriving the key, as supported by our later findings. To the best of our knowledge, the case of Lee metric as distance metric for PUFs has not been considered beforehand. Please note that we are aware of the threat of helper data manipulation attacks [DV]. However, for the presented work, we are interested in discussing more fundamental properties of quantization schemes and ECCs. In addition to that, we assume that access to the helper data is also obstructed by the tamper-evident PUF, i.e., attempts to change the helper data would cause the partial destruction of the PUF as any other physical access to the underlying system. An additional privacy amplification step for the resulting output is always advised but considered out of scope.

3 Tamper-Sensitivity for PUF-based Key Derivation

To further motivate our work, let us briefly discuss an introductory example that hints at the strong need to formalize tamper-sensitivity (TS). When comparing Figure 2a with Figure 2b, then it is striking that the intervals for equidistant quantization are of constant width, whereas the intervals of equiprobable quantization are of unequal width. Consequently, when arbitrarily selecting a value X and subsequently shifting it to the left or right (mimicking an attack), it is easy to see that the magnitude by which X can be shifted *without* changing the obtained symbol varies between these two different approaches. Clearly, the permissible magnitude of the shift without causing $Z^v \neq \hat{Z}^v$ reflects the system's (in)capacity to detect adversarial tampering within $\hat{X}$. Therefore, when a system provides good tamper-sensitivity, it is able to detect even the smallest magnitude changes as a result of the tampering W^{A}.

Here, we deliberately describe the term tamper-sensitivity informally without making any assumptions on the processing of $\hat{X}$ to include processing variants other than those mentioned in this paper, such as [SFIC] or [G]. Furthermore, while we are of the opinion that expressing TS in multiples of the noise standard deviation σ_{N} of the underlying measurement circuit is a reasonable choice for the presented work, it may be too limiting for other models or distributions w.r.t. to the noise. Depending on the type of PUF and specifics of the key derivation scheme, TS should be analyzed for a single measured node as $\mathrm{TS}_{\mathrm{node}}$ or for the whole device as $\mathrm{TS}_{\mathrm{device}}$. For detecting tamper attempts, the property of TS appears to be much more important than effective number of secret bits, as later demonstrated. Based on this generic introduction to tamper-sensitivity, we derive two definitions to more precisely capture a system's capability to detect the tampering W^{A}.

Definition 1 (*max*-TS – Maximum Magnitude Tamper Insensitivity)**.** Defines the *maximum* magnitude of W^{A} that goes *undetected*, i.e., $\max(W^A)$ for which $Z = \hat{Z}$ (or $Z^v = \hat{Z}^v$) still holds. The corresponding notation for a PUF node and device are $\mathrm{TS}^{\max}_{\mathrm{node}}$ and $\mathrm{TS}^{\max}_{\mathrm{device}}$.

max-TS therefore is a *worst*-case scenario from a defender's point of view. Hence, *max*-TS should be minimized to enable better detection of an attacker regardless of the circumstances, i.e., independent for the probability of occurrence of the affected PUF symbols or specifics of the attack. We note that for TS on a device level, either the accumulated per-node TS is considered, or it is normalized by the number v of nodes in

that system to support comparisons across devices with different number of nodes, as included in Table 2. In contrast, we define *min*-TS as follows:

Definition 2 (*min*-TS – Minimum Magnitude Tamper Sensitivity)**.** Defines the *minimum* magnitude of W^{A} that is *detected*, i.e., $\min(W^A)$ for which $Z \neq \hat{Z}$ (or $Z^v \neq \hat{Z}^v$) is achieved. The corresponding notation for a PUF node and device are $\mathrm{TS}^{\min}_{\mathrm{node}}$ and $\mathrm{TS}^{\min}_{\mathrm{device}}$.

It therefore reflects the *best*-case scenario from the defender's point of view to enable earliest detection of an attacker. Within practical limits of applications such as [TS, IOK+18], it is evident that a system performs best when *min*-TS equals *max*-TS and approaches the measurement's noise standard deviation σ_{N}, i.e., the smaller the value for TS is, the better is the sensitivity, i.e., this is equivalent to a small TDT.

These definitions have been formulated such that a hierarchy across different PUF key derivation schemes can be created in a meaningful way, e.g., if *min*-TS(Scheme1) > *max*-TS(Scheme2) is given, then Scheme2 *always* provides a better tamper-sensitivity than Scheme1 and thus, a better detection of attempts to physically tamper with the PUF. Similarly to *min*-entropy as a worst-case scenario for entropy, we are mostly interested in *max*-TS, as it represents the worst-case for the defender. Though not expressed explicitly, preliminary ideas related to these definitions are contained in [IHKS] w.r.t. the quality of the quantization scheme but not to the extent presented in this work.

3.1 Tamper-Sensitivity Equations of Existing Schemes

Let us put the previous definitions to practical use, survey existing schemes, and derive corresponding equations to describe their tamper-sensitivity more analytically. All evaluated schemes have been targeting the scenario of the tamper-evident Coating PUF [TS]. However, specific performance numbers will only be shown later in Section 5 when compared against our scheme that is presented in Section 4.

In the following, we refer to these schemes as profiles to have a semantic difference between the underlying theoretical scheme and its tested instance based on specific parameters. In total, we selected five existing profiles, whereas Profile 1,2, 3, and 4 are based on an equidistant quantization. In case of Profile 1, *only* equidistant quantization is applied *without* subsequent ECC. Profile 2, 3, and 4 then employ an additional ECC after the equidistant quantization. In contrast, Profile 5 is based on an equiprobable quantization and subsequent ECC. These profiles are further detailed hereafter and later compared against Profile 6 which is based on our proposed solution.

TS of Profile 1 based on equidistant quantization *without* ECC [IHKS]: As a baseline, we evaluate the performance of a system that only relies on equidistant quantization without any further processing steps. Following [IHKS], the equidistant quantization is applied to the PUF outputs X. The width Q_{w} of the evenly sized quantization intervals is determined by

$$Q_{\mathrm{w}} = 2 \cdot y \cdot \sigma_{\mathrm{N}} \tag{3}$$

whereas y is a parameter of choice according to the required reliability, i.e., the Confidence Interval (CI) is $[-y \cdot \sigma_{\mathrm{N}}; +y \cdot \sigma_{\mathrm{N}}]$. To obtain m-bit PUF responses, $\mathrm{PDF}(X)$ is divided into $L = 2^m$ intervals of the form $(\mu_X + l \cdot Q_{\mathrm{w}}, \mu_X + (l+1) \cdot Q_{\mathrm{w}}]$ where $l = -L/2, \ldots, -1, 0, 1, \ldots, L/2$. Aligning $l = 0$ and μ_X of the Gaussian distribution leads to the highest entropy output while it is slightly decreased by misalignment depending on the choice of y and the relative shift to μ_X. However, due to symmetry reasons of the equidistant quantization this decrease is well-bounded and therefore a robust scheme.

Figure 3 illustrates the quantization intervals for $m = 4$ and $L = 16$ and an optimal alignment. Each interval is represented by a symbol Q_l in $[0, L-1]$. As the compensated

measurement of the PUF response is non-ideal, i.e., affected by noise of the measurement process, values could move to a different interval compared to the time of enrollment. To counteract this, the offsets between each PUF response X_i and their corresponding interval center are stored as quantization helper data W^*. Upon reconstruction, this offset is applied to the noisy value $\hat{X}_i$ to shift it towards its formerly considered interval center, i.e., $(\hat{X}_i - W_i^* \in Q_{l_i} \rightarrow \hat{Y}_i)$ for $i = 1, \ldots, v$. The full description is included in [IHKS].

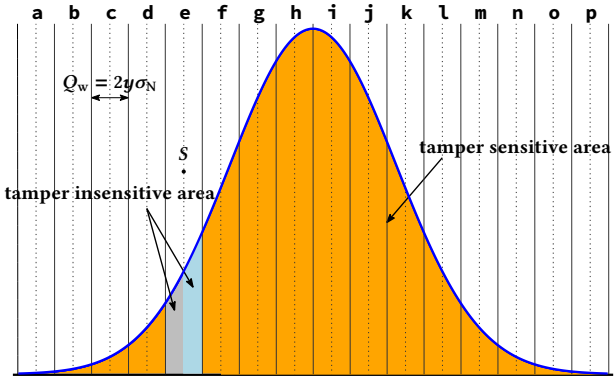


Figure 3: $\mathrm{TS}_{\mathrm{node}}^{\mathrm{max}}$ of Profile 1. Any shift outside of the marked quantization interval causes the detection of a tamper attempt which causes the device to fail (as desired).

When assessing this profile with respect to its tamper-sensitivity, it is best to start with $\mathrm{TS}_{\mathrm{node}}^{\mathrm{max}}$ and visualize its properties as done in Figure 3. Assuming a symbol S at a specific location of the range of values, it is evident that by exceeding its designated quantization interval limits, an erroneous symbol is obtained. The difference between $\mathrm{TS}_{\mathrm{node}}^{\mathrm{max}}(\mathrm{P1})$ and $\mathrm{TS}_{\mathrm{node}}^{\mathrm{min}}(\mathrm{P1})$ is therefore only rooted in a small ϵ that represents the smallest possible resolution step of the underlying measurement circuit. For $\mathrm{TS}_{\mathrm{device}}^{\mathrm{max}}(\mathrm{P1})$, the accumulated tampering that goes undetected on a device-level is therefore the result of $\mathrm{TS}_{\mathrm{node}}^{\mathrm{max}}(\mathrm{P1})$ times the number of nodes v in the system. In contrast, $\mathrm{TS}_{\mathrm{device}}^{\mathrm{min}}(\mathrm{P1})$ is limited by $\mathrm{TS}_{\mathrm{node}}^{\mathrm{min}}(\mathrm{P1})$, i.e., a single erroneous node allows detection of physical tampering. The resulting equations are therefore:

$$\mathrm{TS}_{\mathrm{node}}^{\mathrm{max}}(P1) = Q_w/2 = y \cdot \sigma_N \qquad \mathrm{TS}_{\mathrm{device}}^{\mathrm{max}}(P1) = v \cdot \mathrm{TS}_{\mathrm{node}}^{\mathrm{max}}(P1) \tag{4}$$

$$\mathrm{TS}_{\mathrm{node}}^{\mathrm{min}}(P1) = \mathrm{TS}_{\mathrm{node}}^{\mathrm{max}}(P1) + \epsilon \qquad \mathrm{TS}_{\mathrm{device}}^{\mathrm{min}}(P1) = \mathrm{TS}_{\mathrm{node}}^{\mathrm{min}}(P1) \tag{5}$$

TS of Profile 2 based on Fuzzy Commitment and RS codes [JW, HPKS, IHL$^+$]: Fuzzy commitment is a well-investigated scheme for PUFs and therefore should be considered within the context of this work, too. While the choice of ECC operating on a higher-order alphabet is not limited to RS codes, we chose them to replicate the results of [IHL$^+$]. The basic idea when combining equidistant quantization with an additional ECC is that by making y of Q_w smaller, more entropy can be extracted from the PDF which however does not take into account yet the effects of secrecy leakage by the helper data. At the same time when making y smaller, the failure probability increases and must be counteracted by an ECC which is designated to provide a more flexible approach of counteracting errors when compared to a quantization scheme alone.

Here, we make use of a symbol-based RS code with parameters $\mathrm{RS}(n, t)$, i.e., n as block length in symbols and t as errors to be corrected. RS codes belong to a class of codes called Linear Block Codes. They are represented as $\mathrm{RS}(n, k)$, where k is the number of message symbols and n the block length. A primitive RS Code is defined by a $k \times n$ generator

matrix $\mathrm{G_{RS}}$ as given in Equation (6). RS Codes are Maximum Distance Seperable (MDS), which makes $\mathrm{d_{H|}}_S(\mathrm{RS}(n,k)) = d = n - k + 1$. Hence they can detect and correct up to $d-1$ errors and $t = \lfloor (d-1)/2 \rfloor$ errors respectively.

$$\mathrm{G_{RS}} = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{k-1} & \alpha^{2\cdot(k-1)} & \cdots & \alpha^{(n-1)\cdot(k-1)} \end{pmatrix} \tag{6}$$

where $\alpha \in \mathrm{GF}(2^m)$. The ECC input symbols Y are assumed to be of size $q = |\mathcal{L}|$ and their distance is rated by the Hamming distance $\mathrm{d_{H|}}_S$ which states that any substitution error between $\mathrm{d_{H|}}_S(Y^v, \hat{Y}^v)$ and their symbols, *regardless of their actual distance in the underlying domain of* X, is counted as $\mathrm{d_{H|}}_S(Y, \hat{Y}) = 1$. As an example, $(Y, \hat{Y}) = (a, p)$ yields $\mathrm{d_{H|}}_S = 1$ as shown in Figure 4.

Hence, the scheme operates independently from the actual binary representation of the symbols similar to Profile 1. Consequently, when considering $\mathrm{TS}^{\max}_{\mathrm{node}}(\mathrm{P2})$, the largest magnitude of W^{A} without causing detection may span from the very left to the very right side of the range of values. This corresponds to $q \cdot Q_{\mathrm{w}}$ for $\mathrm{TS}^{\max}_{\mathrm{node}}(\mathrm{P2})$ and indicates already that the detection of W^{A} is rather limited when compared to Profile 1.

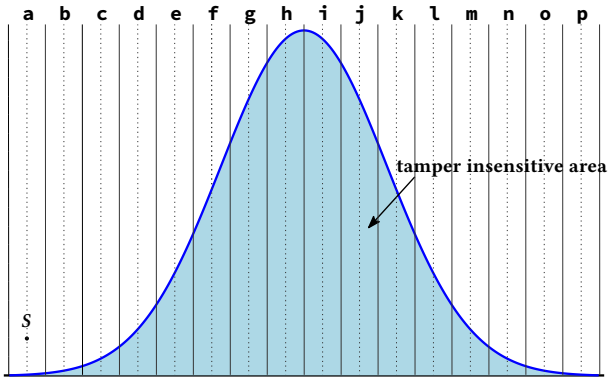


Figure 4: $\mathrm{TS}^{\max}_{\mathrm{node}}$ of Profile 2. Based on a single value X of a node, it is not possible to detect tampering, since any magnitude changes result in $\mathrm{d_{H|}}_S(Y, \hat{Y}) = 1$ due to how Hamming distance is defined for strings of higher-order alphabets.

Since the number of nodes v and symbols derived thereof may not necessarily be equal to the ECC's block length n, it must be divided by a number of segments z for separate processing. This is often owed to the fact that codes with substantial block length are often impractical to implement, especially in hardware implementations. The equation describing $\mathrm{TS}^{\max}_{\mathrm{device}}(\mathrm{P2})$ therefore covers the tampering corrected by the code in its first summand and the remaining tampering that goes undetected by the quantization is contained in the second summand. For $\mathrm{TS}^{\min}_{\mathrm{node}}(\mathrm{P2})$, tampering cannot be detected within a single node for as long as the error threshold t has not been exceeded. To properly define $\mathrm{TS}^{\min}_{\mathrm{device}}(\mathrm{P2})$, we therefore take into account the first summand of $\mathrm{TS}^{\max}_{\mathrm{device}}(\mathrm{P2})$ but then only add $\mathrm{TS}^{\min}_{\mathrm{node}}(\mathrm{P1})$ causing the minimum error to just exceed the scope of the quantization scheme. The resulting equations for TS of P2 are:

$$\mathrm{TS}^{\max}_{\mathrm{node}}(P2) = L \cdot Q_{\mathrm{w}} \qquad \mathrm{TS}^{\max}_{\mathrm{device}}(P2) = z\,t\,\mathrm{TS}^{\max}_{\mathrm{node}}(P2) + (v - z\,t) \cdot \mathrm{TS}^{\max}_{\mathrm{node}}(P1) \tag{7}$$

$$\mathrm{TS}^{\min}_{\mathrm{node}}(P2) = \infty \qquad \mathrm{TS}^{\min}_{\mathrm{device}}(P2) = z\,t\,\mathrm{TS}^{\max}_{\mathrm{node}}(P2) + \mathrm{TS}^{\min}_{\mathrm{node}}(P1) \tag{8}$$

TS of Profile 3 based on Code-Offset and BCH codes [DRS]: Another well-investigated scheme for PUFs is the Code-Offset method. Similar to Profile 2, equidistant quantization is applied. However, this time, the resulting symbols are mapped to bits using a Gray code, i.e., the binary representation of neighboring quantization intervals differs by a hamming distance of 1 only, as it was done also in [TS] for equiprobable quantization, as later considered in Profile 5. After this bit mapping to Gray coded symbols, a BCH code is applied. BCH codes can also be described as binary RS codes, i.e., they are represented as $\mathrm{BCH}[n,k,d]_{\mathrm{GF}(2^m)}$. Correspondingly, the distance between codewords is counted by the Hamming distance $\mathrm{d}_{\mathrm{H}|2}$.

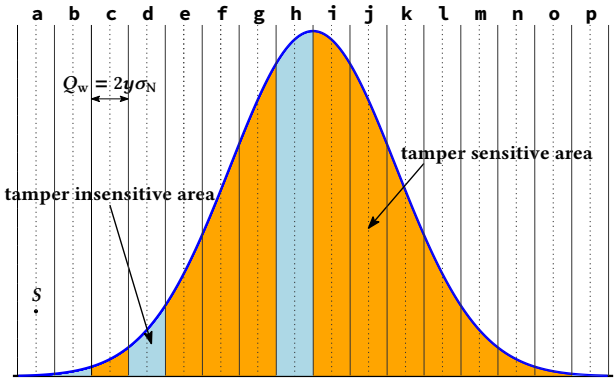


Figure 5: $\mathrm{TS}^{\max}_{\mathrm{node}}$ of Profile 3. Please note that for Gray encoded symbols, the resulting distance $\mathrm{d}_{\mathrm{H}|2}(a,p)=1$, due to how a Gray code is typically constructed.

The basic idea of this scheme is as follows: Errors close to the designated value result in a small Hamming distance, while a larger shift will increase the Hamming distance. We observe that $L=2^m$, i.e., m as number of bits to encode the intervals. Since $m<L$, it follows that there exists only one case of the codebook where $\mathrm{d}_{\mathrm{H}|2}$ per node is maximized, i.e., $\mathrm{d}_{\mathrm{H}|2}(Y,\hat{Y})=m$. This is the case when the all null bit sequence derived from a node is flipped to the all one bit sequence. In all other cases, $\mathrm{d}_{\mathrm{H}|2}(Y,\hat{Y})\leq m-1$ which degrades the tamper-sensitivity of the device. Even worse, some very extreme magnitude shifts may result in only $\mathrm{d}_{\mathrm{H}|2}(Y,\hat{Y})=1$ due to how a Gray code is constructed. For the example given in Figure 5, when assuming a Gray code as follows: $(a\leftarrow 0000)$, $(b\leftarrow 0001)$, $(c\leftarrow 0011)$, $\ldots$, $(p\leftarrow 1000)$, then the largest possible shift while ensuring a Hamming distance of 1 is from the symbol a to the symbol p. Correspondingly, *max*-TS for this profile results in

$$\begin{aligned}\mathrm{TS}^{\max}_{\mathrm{node}}(P3)&=L\cdot Q_{\mathrm{w}}\\ \mathrm{TS}^{\max}_{\mathrm{device}}(P3)&=z\,t\,\mathrm{TS}^{\max}_{\mathrm{node}}(P3)+(v-z\,t)\cdot\mathrm{TS}^{\max}_{\mathrm{node}}(P1)\end{aligned}\tag{9}$$

To write a closed form of $\mathrm{TS}^{\min}_{\mathrm{node}}(\mathrm{P3})$ and $\mathrm{TS}^{\min}_{\mathrm{device}}(\mathrm{P3})$, we assume that the attacker can divide and distribute W^{A} such that indeed only the smallest detectable change in $\mathrm{d}_{\mathrm{H}|2}$ per node occurs. For equiprobable quantization this is a symbol residing in any interval with width Q_{w} and shifting to its directly neighboring intervals, thereby causing a single bit substitution error. When $t>1$ the ECC is capable of correcting more bits, then multiple nodes with a single bit error within a segment z could be corrected, or larger magnitude shifts within a node (which is not desired with regard to tamper-sensitivity). However, to adhere to the definition of *min*-TS, we assume that for larger t, indeed t-times the smallest detectable change occured. The resulting equations are therefore:

$$\begin{aligned} \mathrm{TS}^{\min}_{\mathrm{node}}(P3) &= 3 \cdot Q_{\mathrm{w}}/2 + \epsilon \qquad \text{iff } t = 1 \\ \mathrm{TS}^{\min}_{\mathrm{device}}(P3) &= z\,t\,\mathrm{TS}^{\min}_{\mathrm{node}}(P3) + \mathrm{TS}^{\min}_{\mathrm{node}}(P1) \end{aligned} \tag{10}$$

TS of Profile 4 based on VT-like codes [IHL$^+$]: This profile again is based on an equidistant quantization but this time with a variable-length mapping of the symbols Y to bits, as described in [IHL$^+$]. The corresponding code is a VT-like code denoted as $\mathrm{VT}(\cdot, t)$ with t as number of errors in $\mathrm{d_{Lev}}$. Due to the limitations of VT-codes, $t = 1$ always, as multiple insertion/deletion errors can only be corrected when considering multiple segments z.

VT-like codes are designed using *Levenstein Distance* metric. Each derived symbol Y corresponding to a quantization interval is bit mapped to a variable number of bits. Since these bit maps should be uniquely decodable, they are generated using a binary tree, while ensuring $\mathrm{d_{Lev}} = 1$ between neighboring intervals. For a PUF device with v nodes, z number of VT-like code segments can be generated. Since each segment can correct only 1 symbol error, the total number of correctable symbol errors is z. The systematic code construction is described in Equation (11) following the notation of [IHL$^+$].

$$\mathrm{C_{VT}} := \left\{ (b_1, b_2, ..., b_{\mathrm{m}}, p_1, ..., p_{\mathrm{r}}) : \sum_{i=1}^{m} i \cdot b_{\mathrm{i}} + \sum_{j=1}^{r} 2^{j-1} \cdot p_{\mathrm{j}} \equiv 0 \pmod{2m+1} \right\} \tag{11}$$

$$\mathrm{P} = 2m + 1 - \left(\sum_{i=1}^{m} i \cdot b_{\mathrm{i}} \pmod{2m+1} \right) \tag{12}$$

where $(b_1, b_2, ..., b_{\mathrm{m}})$ is the bit map of $(y_1, y_2, ..., y_{\mathrm{v/z}})$ PUF nodes and $(p_1, ..., p_{\mathrm{r}})$ is the binary representation of P given by Equation (12).

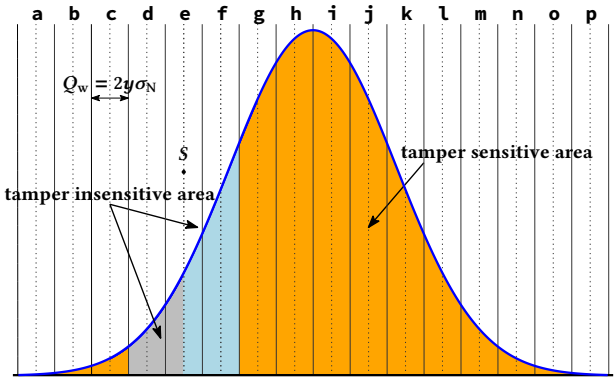


Figure 6: $\mathrm{TS}^{\min}_{\mathrm{node}}$ of Profile 4.

When analyzing the tamper-sensitivity of this profile, it is evident that writing a closed form for $\mathrm{TS}^{\max}_{\mathrm{node}}(\mathrm{P4})$ and $\mathrm{TS}^{\max}_{\mathrm{device}}(\mathrm{P4})$ is difficult, as it depends on the number of quantization intervals and the codebook used to create the variable-length bit mapping[2]. This statement is based on the observation that $\mathrm{d_{Lev}}(Y, \hat{Y}) = 1$ is ensured for directly neighboring intervals but larger magnitude changes may still result in distance $\mathrm{d_{Lev}} = 1$, i.e., the attacker may be even encouraged to cause larger magnitude changes that would still be accounted for by the error-correcting capability of the code. We therefore directly

[2]For the specific case later considered: $\mathrm{TS}^{\max}_{\mathrm{node}}(P4) = 6 \cdot Q_{\mathrm{w}} + Q_{\mathrm{w}}/2$ for 12 intervals; $\mathrm{TS}^{\max}_{\mathrm{node}}(P4) = 10 \cdot Q_{\mathrm{w}} + Q_{\mathrm{w}}/2$ for 14 intervals; and on a device-level: $\mathrm{TS}^{\max}_{\mathrm{device}}(P4) = z\,t \cdot \mathrm{TS}^{\max}_{\mathrm{node}}(P4) + (v - z\,t) \cdot \mathrm{TS}^{\max}_{\mathrm{node}}(P1)$

compute *max*-TS values for the parameters later considered by using the codebooks provided by the authors of [IHL+] (cf. Appendix A). In contrast, stating *min*-TS equations is straightforward and visualized in Figure 6. The corresponding equations are

$$\mathrm{TS}^{\min}_{\mathrm{node}}(P4) = 3\,Q_{\mathrm{w}}/2 + \epsilon \qquad \mathrm{TS}^{\min}_{\mathrm{device}}(P4) = \mathrm{TS}^{\min}_{\mathrm{node}}(P4) \tag{13}$$

owed to the fact that the minimum error to detect is the one just exceeding the error-correcting capability of the VT-like code. Similarly to Profile 1, the overall $\mathrm{TS}^{\min}_{\mathrm{device}}(\mathrm{P4})$ is again the same as $\mathrm{TS}^{\min}_{\mathrm{node}}(\mathrm{P4})$, i.e., a single erroneous node triggers the tamper-detection which is a beneficial behavior for improved tamper-sensitivity.

TS of Profile 5 based on Equiprobable Quantization and BCH-based Code-Offset [TS]: Unlike before, we make use of an equiprobable quantization and refer to [TS] for its formal description. As illustrated in Figure 2b, this approach is characterized by its innermost intervals of width $Q_{\min}$ and outermost intervals of width $Q_{\max}$. As described in Section 2.2, the symbols are mapped to a binary representation using a Gray code. A BCH(n,t) code is applied to the resulting output, whereas both n and t are in bits.

One of the challenges for this profile is defining the outermost intervals properly when considering a practical implementation, as equal probability of intervals needs to be ensured also for the outermost intervals that are however limited by the measurement range of the underlying implementation. Values outside of the measurement range cannot be used for tamper-detection. Excluding values with probability of occurrence $< 0.1\%$ per node balances the engineering effort for the measurement circuit with the expected excess during production, assuming that devices with values outside of the measurement range are discarded. Hence, at some point in the range of X, the tails of the PDF need to be cut off. If this would not be done, then tamper-sensitivity would not be bounded in the outermost intervals of the equiprobable quantization. This would no longer represent the targeted practical scenario and thus, result in an unfair comparison that we want to avoid. The same has been done in [IHKS] or [WHGS].

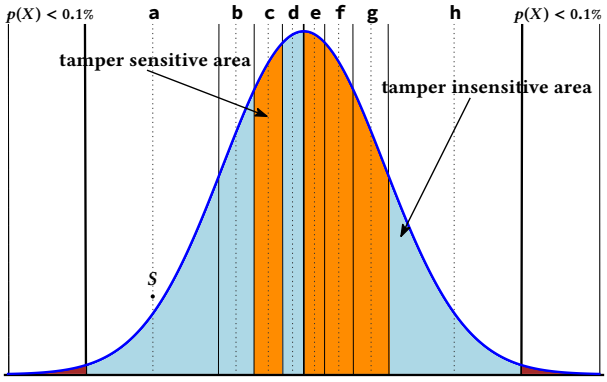


Figure 7: $\mathrm{TS}^{\max}_{\mathrm{node}}$ of Profile 5 for the symbol S as indicated. Based on the Gray code bit mapping as illustrated in Figure 2b and as further detailed in Appendix A.

Regarding the tamper-sensitivity of this profile, we observe similarly to P3 that the specifics of the Gray code significantly affect the tamper-sensitivity. For example, for the scenario presented in [TS], a *shift* from the left outermost interval to the right would only result in distance 1, as illustrated in Figure 7 when assuming the bit mapping of Figure 2b. *max*-TS for this profile therefore results in

$$\begin{aligned} \mathrm{TS}^{\max}_{\mathrm{node}}(P5) &= \sum_{i=1}^{L} \mathrm{width}(Q_i) \\ \mathrm{TS}^{\max}_{\mathrm{device}}(P5) &= z\,t\,\mathrm{TS}^{\max}_{\mathrm{node}}(P5) + (v - z\,t)\cdot Q_{\max}/2 \end{aligned} \tag{14}$$

To write a closed form of $\mathrm{TS}^{\min}_{\mathrm{node}}(\mathrm{P5})$ and $\mathrm{TS}^{\min}_{\mathrm{device}}(\mathrm{P5})$, we again assume that the attacker can divide and distribute W^{A} such that indeed only the smallest detectable change in $\mathrm{d}_{\mathrm{H|2}}$ per node occurs. For equiprobable quantization this is a symbol residing in $Q_{\min}$ and shifting to its neighboring intervals. When $t > 1$, then multiple nodes could be corrected or larger magnitude shifts within a node. To adhere to the definition of *min*-TS, we assume that for larger t, indeed t-times the smallest detectable change occured. The resulting equations are therefore:

$$\begin{aligned} \mathrm{TS}^{\min}_{\mathrm{node}}(P5) &= 3\cdot Q_{\min}/2 + \epsilon \qquad \text{iff } t = 1 \\ \mathrm{TS}^{\min}_{\mathrm{device}}(P5) &= z\,t\,\mathrm{TS}^{\min}_{\mathrm{node}}(P5) + Q_{\min}/2 + \epsilon \end{aligned} \tag{15}$$

Regarding the fairness of comparison and the design trade-off made w.r.t. $Q_{\max}$, i.e., where to cut off the range of values, we point out that by defining *min*-TS as given, it is independent from the size of the outermost interval. Hence, it only affects *max*-TS whereas excluding more values would make $Q_{\max}$ smaller but increase excess during the manufacturing process, thereby reducing the yield.

3.2 Discussion and Comments Regarding Tamper-Sensitivity

All presented TS equations have in common that they describe a noise-free scenario, as motivated in Section 2.1 for analysis purposes only. This simplifies the equations without affecting their accuracy in describing the fundamental TS property of the scheme. Moreover, we neglect the challenges that arise when trying to define TS for the outermost intervals of a specific profile, i.e., independent of the actual measurement range of the PDF that could be covered and the number of quantization intervals to sample it. We assume that TS is not affected by these practical constraints and instead is purely based on the properties of the underlying scheme.

We also note that excluding values with probability less than 0.1% per node for Profile 5 would result in a yield reduction of 12% overall for a device with 128 nodes. If in contrast less values would be excluded to improve the yield, then at the same time TS in the outermost intervals would be degraded. Since we primarily evaluate TS, we deliberately chose a poor yield as this improves tamper-sensitivity in the outermost interval. We emphasize for Profiles 1,2,3, and 4 that choosing the cut-off for equidistant quantization does not change the tamper-sensitivity behavior in the outermost intervals, as they are always of a fixed width (later, the same applies to Profile 6). Since working with *min*-entropy, changing the number of equidistant quantization intervals while keeping their width constant does not increase the extracted entropy. In general, for analysis of the conceptual schemes only, we suppose that the range of the measurement circuit is not bounded. In a practical setting, both enrollment and reconstruction would be done with a measurement circuit that provides a bounded range. However, since the anticipated yield reduction is independent from the TS properties in case of equidistant quantization, and also heavily depends on the manufacturers business model, i.e., factors outside of the technical scope of this paper, we do not further take this aspect into account.

We point out that our definition of TS assumes unidirectional *shifts*, i.e., a change in value cannot be in both directions at the same time. This is of particular relevance for Profile 3, 4, and 5, where a shift may move values over intervals that are considered

tamper-sensitive. Hence, *not* the tamper-sensitive area of a PDF is taken into account but indeed the magnitude of the shifts only. Since Profile 5 deviates already at the point of quantization from the other profiles, neither $\mathrm{TS}_{\mathrm{node}}^{\mathrm{min}}$ nor $\mathrm{TS}_{\mathrm{node}}^{\mathrm{max}}$ will reflect the perceived tamper-sensitivity in a practical setting as it will be based on the *average* tamper-sensitivity that takes into account the probability of occurrence of an affected quantization interval, i.e., it would be necessary to weigh the tamper-sensitivity per interval by its probability of occurrence. However, *min*-TS and *max*-TS already provide a quality assessment to sufficiently compare Profile 5 against the other profiles.

As can be derived from the given equations, all schemes behave differently when considering $\mathrm{TS}_{\mathrm{node}}^{\mathrm{max}}$ and $\mathrm{TS}_{\mathrm{device}}^{\mathrm{max}}$. This already supports the argument that a property is being addressed that otherwise cannot be captured by entropy or failure rate. Please note that while some of the given equations appear highly similar, e.g., $\mathrm{TS}_{\mathrm{node}}^{\mathrm{min}}$ of Profiles 1, 3, 4, and 5, their actual value will still be different when considered under a specific set of parameters. The interested reader may already proceed to Table 2 to see the resulting numbers for the tested profiles. Correspondingly will the visual appearance of the presented Figures for the actual parameters be different, e.g., smaller but more intervals.

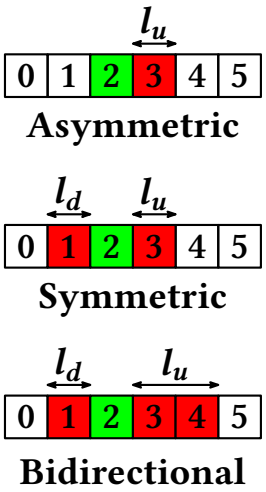
Late Tamper Evaluation. Our work focuses on improving the combination of quantization and ECC *without* additional processing steps. Alternatively, it may be possible for some profiles to further improve tamper detection by studying the magnitude of errors *after successful decoding* was done, i.e., by computing the Euclidian distance $\mathrm{d_E}(\hat{Z}, \hat{X})$ and validating that the result is of a reasonable magnitude, e.g., by requiring $\mathrm{d_E}(\hat{Z}, \hat{X}) \leq \mathrm{TDT}$, e.g., $\mathrm{TDT} = 3 \cdot Q_{\mathrm{w}}/2 = 3 \cdot y \cdot \sigma_{\mathrm{N}}$ for Profile 4. By following this approach for Profile 4, it is possible to limit the error magnitude to $\mathrm{TS}_{\mathrm{node}}^{\mathrm{min}}$(Profile 4). This is possible since only one error per segment z is covered by the scheme. For other Profiles though, such as Profile 3 and Profile 5, this quickly leads to inconsistencies in how errors are treated. This argument is based on the observation that within a block of length n (in bits), up to t errors (in bits) are corrected. Assuming that m bits per node are derived and $t > m$ (which is the case for the practical scenarios considered), then it is becoming increasingly difficult to formulate a valid late tamper evaluation approach, since the late tamper evaluation will impede with how the ECC operates. Hence, even if such an approach can be successfully applied to any of the existing profiles, the obtained result will still not exceed the *min*-TS level due to how it has been defined. This is in addition to the potential security threat of first reconstructing the valid secret, before discarding it based on the result of the late tamper evaluation. To the best of the author's knowledge, there are no other publications discussing the specifics of such a late tamper evaluation.

In the next Section, we present our approach based on Limited Magnitude Codes (LMC) in combination with equidistant quantization which is then named Profile 6 as part of the comparison. This enables $\mathrm{TS}_{\mathrm{node}}^{\mathrm{min}}$ to be equal to $\mathrm{TS}_{\mathrm{node}}^{\mathrm{max}}$ and $\mathrm{TS}_{\mathrm{device}}^{\mathrm{min}}$, as it is the case for Profile 1 but at the same time, be almost twice as tamper-sensitive on a device-level in addition to extracting more entropy.

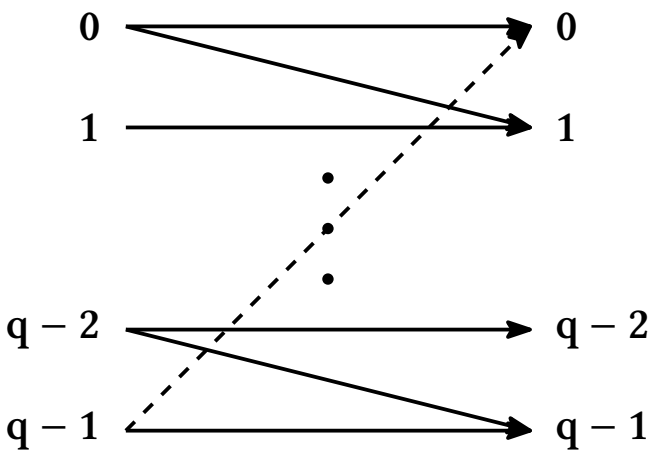
4 Limited Magnitude Codes (LMC)

As indicated by our analysis of the previous work, specifically of Profile 3, 4 and 5, mapping higher-order alphabet symbols to an alphabet of lower degree diminishes tamper-sensitivity by causing an unevenly spread TS in the codebook. However, also building upon inappropriate distance metrics such as Hamming distance over symbols, as done in Profile 2, degrades tamper-sensitivity, as the distance $\mathrm{d_E}(X, \hat{X})$ is not well reflected by $\mathrm{d_{H|}}_S(Y, \hat{Y})$. To solve these problems, we model the outcome of the equidistant quantization as a q-ary channel as depicted in Figure 8b, i.e., we continue operating on the symbols directly. In contrast to previous works, we rate errors in this channel by the *Lee* metric $\mathrm{d_{Lee}}$, i.e., symbols of neighboring intervals will have a distance of 1 whereas symbols of larger

distance l will have distance l. This is also called the *magnitude*. Different possible types of magnitude errors are illustrated in Figure 8a. These are classified as *asymmetric* when unidirectional, *symmetric* when of equal magnitude in either direction, or *bidirectional* when in either direction but of unequal magnitude.



(a) LMC error types for $q = 5$ and the designated symbol 2.



(b) q-ary asymmetric channel model for $l = 1$ between symbols. Non-wrap around channel (solid lines only) and wrap-around channel (solid lines and dashed line as wrap-around).

Figure 8: LMC error types and q-ary channel model.

Elarief et al. [EB] first proposed a code to correct all asymmetric and symmetric errors of limited magnitude in a q-ary channel. While the code proposed by [EB] corrects *all* magnitude errors, it does not allow to limit the number of magnitude errors corrected by the ECC which is the necessary degree of design freedom we need. To mitigate this shortcoming, Myeongwoon et al. [JL] proposed a modified version of this code called Limited-Magnitude Error Correction Code (LMC). This is based on an RS Encode/Decode step that is additionally introduced to limit the number of correctable errors as later described. Hence, this can be considered as a concatenated code construction of LMC and RS codes, whereas we are not limited to RS codes but could have selected any other code operating on higher-order alphabet symbols. Although the new code by [JL] was intended for bidirectional errors, it is equally applicable to asymmetric and symmetric errors.

The error correction capability of these codes is as follows (cf. Figure 8a): In Asymmetric LMC (A-LMC), a symbol is correctable if the possible error occurs in only one direction. For example, if the symbol is *2* then in A-LMC ($l_\mathrm{u} = 1$) the symbol is corrected only if it changes to *3* (error $= +1$). If the symbol changes to any other value, it is not corrected. Similarly for Symmetric LMC (S-LMC), the error magnitude can be ± 1 i.e. $l_\mathrm{u} = |l_\mathrm{d}| = 1$. This implies that even if symbol *2* becomes *1*, it is corrected. Bidirectional LMC (B-LMC) is a generic case of S-LMC where $|l_\mathrm{u}| \neq |l_\mathrm{d}|$.

These error types can be considered within the scope of two different q-ary channel models. They are called wrap-around and non-wrap-around channel. In Figure 8b the wrap-around is indicated by a dashed line, whereas all other lines are solid and represent the only valid transitions for the non-wrap-around channel. Hence, for the wrap-around channel, $\mathrm{d}_\mathrm{Lee}(q-1, 0) = 1$, whereas for the non-wrap-around channel $\mathrm{d}_\mathrm{Lee}(q-1, 0) = q-1$. Since the underlying application is based on a physical measurement process, the wrap-around is not desirable and counteracts the aspect of tamper-sensitivity. Therefore, to minimize $\mathrm{TS}_\mathrm{node}^\mathrm{max}$ and best reflect $\mathrm{d}_\mathrm{E}(X, \hat{X})$ in the quantized symbols $\hat{Y}^v$, we only make use of the non-wrap-around channel model. The Lee metric in the non-wrap around channel is

sometimes also termed *Manhattan* distance d_{Man}.

For encoding and decoding, the corresponding steps are listed in Algorithm 1 and Algorithm 2 that are described by [JL]. The parameters of an LMC are $q\prime$, q, p and t. $q\prime$ represents the number of values a symbol can take under the influence of an error, while still being within the LMC boundary. q represents the number of quantization intervals. p is the RS code field size GF(p) and t is the error correction capability of RS code. While constructing any LMC, Equation (16) must always hold.

$$q\prime = l_u + |l_d| + 1 \quad \text{and} \quad q\prime \leq q \leq p \tag{16}$$

The encoding and decoding algorithm is complemented by Algorithm 3 which is instantiated by both LMC Encode and Decode and helps translating an array of elements from one base to another, especially for the description presented here, assuming that q' is a power of 2, allowing for a very efficient implemenation as demonstrated by the examples in the appendix.

Algorithm 1: LMC Encode

Data: $Y = [y_1, y_2, ...y_v] \in [0, q-1]$
Result: $Z = [z_1, z_2, ...z_v] \in [0, q-1]$, W
`/* Step 1: Calculate remainder of` $\frac{Y}{q\prime}$ `*/`
1 $\eta = Y \pmod{q\prime}$
`/* Step 2: Generate p-ary message symbols using` η `and encode it using RS(`n`,`t`) encoder. */`
2 $\eta_p = \text{baseChange}(\eta, q\prime, p)$
3 $C = \text{RS}_{\text{Enc}}(\eta_p, n, t)$
`/* Step 3: Convert` $2t$ `p-ary parity symbols to q-ary symbols */`
4 $W = \text{baseChange}(C[n-2t+1:n], p, q)$
`/* Step 4: Set output` Z `*/`
5 $Z = Y$

The algorithms for encoding and deconding can be used for A-LMC and S-LMC as well, by changing $q\prime$ as in Equation (17). If we correct t times a p-ary error, then the maximum number of $q\prime$-ary errors potentially corrected by LMC is given by $t_{\max}$ as defined in Equation 18. Since the minimum number of errors corrected is t, we use t as the number of errors corrected by LMC for notation purposes and also computation of the reliability. However, for *max*-TS, we indeed use $t_{\max}$. This could be even further improved by making use of the early decoding termination, as introduced in the subsequent paragraph.

$$q\prime = \begin{cases} l_u + |l_d| + 1, & \text{B-LMC} \\ 2l_u + 1, & \text{S-LMC} \\ l_u + 1, & \text{A-LMC} \end{cases} \tag{17}$$

$$t_{\max} = \frac{t \cdot \log_2(p)}{\log_2(q')} \tag{18}$$

Early Decoding Termination: We introduce an additional check on the number of non-zero elements in ε'' to limit the maximum number of $q\prime$-ary errors that get corrected. If the number exceeds the chosen threshold t, then a decoding failure can be triggered. (cf. lines 13 to 16 of Algorithm 2). Once a decoding error occurs, the device should enter a permanent failure mode from which recovery is difficult, e.g., by blowing fuses or zeroization of helper data. This is required to not introduce an obvious timing side-channel in the decoding process and is within the principles of tamper-detection and response.

Algorithm 2: LMC Decode

Data: $\hat{Y} = [\hat{y}_1, \hat{y}_2, ...\hat{y}_v] \in [0, q-1], W, e \in \{\text{TRUE}, \text{FALSE}\}$
Result: $\hat{Z} = [\hat{z}_1, \hat{z}_2, ...\hat{z}_v] \in [0, q-1]$
/* Step 1: Calculate remainder of $\frac{\hat{Y}}{q\prime}$ */
$\varphi = \hat{Y} \pmod{q\prime}$
/* Step 2: Convert φ and W to p-ary and form a codeword. */
$\varphi_p = \text{baseChange}(\varphi, q\prime, p)$
$P = \text{baseChange}(W, q, p)$
$C' = [\, \varphi_p \,||\, P \,]$
/* Step 3: Correct the codeword using RS(n,t) decoder. */
$\hat{C} = \text{RS}_{\text{Dec}}(C', n, t)$
/* Step 4: Convert the message part of $\hat{C}$ to $q\prime$-ary and estimate the error */
$\varphi\prime = \text{baseChange}(\hat{C}[1 : n-2t], p, q\prime)$
$\varepsilon\prime = \varphi - \varphi\prime = [\varepsilon_1\prime, \varepsilon_2\prime ... \varepsilon_v\prime]$
/* Step 5: Refine error to lie in $[l_\text{d} \quad l_\text{u}]$ bound */
for $i \leftarrow 1$ **to** v **do**
  if $\varepsilon'_i < l_\text{d}$ **then**
    $\varepsilon''_i = \varepsilon'_i + q\prime$
  else if $\varepsilon'(i) > l_\text{u}$ **then**
    $\varepsilon''_i = \varepsilon'_i - q\prime$
  if $\varepsilon''_i \neq 0$ **then**
    count = count + 1 // required only for Early Termination
/* Optional: Early Decoding Termination */
if $e ==$ TRUE & count $> t$ **then**
  return
/* Step 6: Subtract ϵ'' from $\hat{Y}$ to get the corrected output */
$\hat{Z} = \hat{Y} - \epsilon''$

Algorithm 3: LMC baseChange

Data: $D^\text{In} = [d_1, d_2, ...d_n]$, baseIn, baseOut
Result: $D^\text{Out} = [d_1, d_2, ...d_m]$
baseInBits $= \lceil \log_2(\text{baseIn}) \rceil$
baseOutBits $= \lceil \log_2(\text{baseOut}) \rceil$
/* Step 1: Represent each array element of D^In in binary using dec2bin() */
for $i \leftarrow 1$ **to** n **do**
  $D_b[i \cdot \text{baseInBits} : (i+1) \cdot \text{baseInBits}] = \text{dec2bin}(D^\text{In}[i], \text{baseInBits})$
/* Step 2: Estimate number of elements in D^Out */
$m = \lceil n \cdot \text{baseInBits} / \text{baseOutBits} \rceil$
/* Step 3: Combine each baseOutBits elements of D_b to form one symbol using bin2dec() */
for $i \leftarrow 1$ **to** m **do**
  $D^\text{Out}[i] = \text{bin2dec}(D_b[i \cdot \text{baseOutBits} : (i+1) \cdot \text{baseOutBits}], \text{baseOutBits})$

Secrecy Leakage by Helper Data: The leakage caused by LMC helper data W is upper bounded using Equation 19, since it is essentially a Code-Offset construction where only the parity is stored. Here, $P = z \cdot 2\,t \cdot \log_2(p)$ is the total number of parity bits P generated for z segments of LMCs, based on the RS code operating in the p-ary domain.

$$\mathrm{I}(X^v; W) = \lceil P \rceil = \lceil z \cdot 2\,t \cdot \log_2(p) \rceil \,\mathrm{bit} \tag{19}$$

The leakage calculation for the first entry of Profile 6 in Table 1 is provided as an example in the following. First, we compute the secrecy leakage.

$$\mathrm{I}(X^v; W) = \lceil z \cdot 2\,t \cdot \log_2(p) \rceil = \lceil 1 \cdot 2 \cdot 10 \cdot \log_2(64) \rceil = 120\,\mathrm{bit}$$

Concerning the *min*-entropy that is extracted on average from a device, we consider each node with parameter $y = 2.1$ for the equidistant quantization which leads to a *min*-entropy $\tilde{\mathrm{H}}_\infty(Y)$ of 3.4325 bit per node, resulting in an overall *min*-entropy for a device $\tilde{\mathrm{H}}_\infty(Y^v)$ with $v = 128$ nodes of

$$\tilde{\mathrm{H}}_\infty(Y^v) = v \cdot \tilde{\mathrm{H}}_\infty(Y) = 3.4325 \cdot 128 = 439.36\,\mathrm{bit}$$

Hence, the effective number of secret bit, i.e., when accounting for the previously computed helper data leakage, is

$$\mathrm{H}_\infty^{\mathrm{eff}} = \tilde{\mathrm{H}}_\infty(Y^v) - \mathrm{I}(Y^v; W) = 439.36 - 120 \approx 319\,\mathrm{bit}$$

Failure Probability Computation: Based on the presented LMC properties, decoding fails if one of the following conditions is met:

1. The magnitude of error ε exceeds $[l_\mathrm{d}\ l_\mathrm{u}]$ of the LMC
2. The number of p-ary errors is greater than t, i.e., too many magnitude errors in total

To provide a generic description of the failure probability, let r parts constitute a symbol, i.e., the number of unique digits to represent the symbol (radix). Let $\mathrm{P}_{\mathrm{part}}$ be the error probability of one part and the symbol error probability be $\mathrm{P}_{\mathrm{symb}}$. Then the error probability of a symbol is computed as

$$\mathrm{P}_{\mathrm{symb}}(r, \mathrm{P}_{\mathrm{part}}) = \sum_{i=1}^{i=r} \binom{r}{i} \cdot {\mathrm{P}_{\mathrm{part}}}^i \cdot (1 - \mathrm{P}_{\mathrm{part}})^{r-i} \tag{20}$$

We know that

$$\begin{aligned} \mathrm{P}_{\mathrm{symb}} + \binom{r}{0} \cdot {\mathrm{P}_{\mathrm{part}}}^0 \cdot (1 - \mathrm{P}_{\mathrm{part}})^{r-0} &= 1 \\ \mathrm{P}_{\mathrm{symb}} + (1 - \mathrm{P}_{\mathrm{part}})^r &= 1 \\ \implies \mathrm{P}_{\mathrm{part}}(r, \mathrm{P}_{\mathrm{symb}}) &= 1 - (1 - \mathrm{P}_{\mathrm{symb}})^{1/r} \end{aligned} \tag{21}$$

If the incorporated ECC corrects up to t errors then error probability after ECC is given by Equation (22). Where $\mathrm{P} = \mathrm{P}_{\mathrm{symb}}$ for RS code and $\mathrm{P} = \mathrm{P}_{\mathrm{bit}}$ for BCH code. P_e is the error probability of one block of RS/BCH code.

$$\mathrm{P}_\mathrm{e}(n, t, \mathrm{P}) = \sum_{i=t+1}^{i=n} \binom{n}{i} \cdot \mathrm{P}^i \cdot (1 - \mathrm{P})^{n-i} \tag{22}$$

For the error probability calculation of LMC, we assume that after LMC decode, the q-ary symbol error probability ($\mathrm{P}_\mathrm{e}(\mathrm{Z}^\mathrm{v})$) depends only on q'-ary errors. The errors of magnitude

$> q'$ are not used in the calculation since this is considered as tampering. Based on the previous equations, Algorithm 4 provides the approach on how to compute the error probabilities. Please note that it provides an upper bound for the failure probability for LMC cases where $\log_2(q)/\log_2(q')$ and $log_2(p)/log_2(q')$ are not integers. The resulting performance numbers for the considered parameters are presented in Table 1, alongside all other profiles. In the following, the tamper-sensitivity of LMC is analyzed.

Algorithm 4: LMC Error Probability

Data: $\mathrm{P_e(Y)}, q', q, p, z$
Result: $\mathrm{P_e(Z^v)}$
// Step 1: Calculate q'-ary symbol error probability before RS Decoder using Equation (21)
1 $\mathrm{P_{q'\_symb}} = \mathrm{P_{part}}(\lceil \log_2(q)/\log_2(q') \rceil, \mathrm{P_e(Y)})$
// Step 2: Calculate p-ary symbol error probability before RS Decoder using Equation (20)
2 $\mathrm{P_{p\_symb}} = \mathrm{P_{symb}}(\lceil \log_2(p)/\log_2(q') \rceil, \mathrm{P_{q'\_symb}})$
// Step 3: Calculate p-ary block error probability after RS Decoder using Equation (22)
3 $\mathrm{P_{e\_block\_rs}} = \mathrm{P_e}(n, t, \mathrm{P_{p\_symb}})$
// Step 4: Calculate p-ary symbol error probability after RS Decoder using Equation (21)
4 $\mathrm{P_{p\_symb\_rs}} = \mathrm{P_{part}}(n, \mathrm{P_{e\_block\_rs}})$
/* Step 5: Calculate q'-ary symbol error probability after LMC Decoder using Equation (21) */
5 $\mathrm{P_e(Z)} = \mathrm{P_{part}}(\lceil \log_2(p)/\log_2(q') \rceil, \mathrm{P_{p\_symb\_rs}})$
/* Step 6: Calculate q-ary block error probability after LMC Decoder using Equation (20). Note, there are $\lceil k \cdot \log_2(p)/\log_2(q') \rceil$ q-ary symbols in 1 block of LMC */
6 $\mathrm{P_e(Z_z)} = \mathrm{P_{symb}}(\lceil k \cdot \log_2(p)/\log_2(q') \rceil, \mathrm{P_e(Z)})$
/* Step 7: Calculate q-ary device error probability after LMC Decoder using Equation (20). There are z blocks of LMC per device. */
7 $\mathrm{P_e(Z^v)} = \mathrm{P_{symb}}(z, \mathrm{P_e(Z_z)})$

TS of Profile 6 based on Equidistant Quantization and LMC: Following the previous description, LMC corrects t errors within the $[l_d \quad l_u]$ boundary. Hence $\mathrm{TS^{max}_{node}(P6)}$ is defined using Equation 23. Its first summand is based on the error correction capability of the LMC and the second summand caused by the equidistant quantization. Hence, to cause detection, an additional ϵ is required for $\mathrm{TS^{min}_{node}(P6)}$. Since LMC decoding fails even if the number of errors is less than t but the magnitude exceeds $[l_d \quad l_u]$, $\mathrm{TS^{min}_{device}(P6)}$ is equivalent $\mathrm{TS^{min}_{node}(P6)}$. This already indicates a significant advantage over the other profiles discussed earlier. Calculating $\mathrm{TS^{max}_{device}(P6)}$ then follows similar principles of the other ECC-based profiles, i.e., if the block length n does not match the input length of symbols v, then multiple segments z must be created.

$$\mathrm{TS^{max}_{node}}(P6) = \max(l_\mathrm{u}, |l_\mathrm{d}|) \cdot Q_\mathrm{w} + \mathrm{TS^{max}_{node}}(P1) \tag{23}$$

$$\mathrm{TS^{max}_{device}}(P6) = z\, t_{\max} \cdot \mathrm{TS^{max}_{node}}(P6) + (v - z\, t_{\max}) \cdot \mathrm{TS^{max}_{node}}(P1) \tag{24}$$

$$\mathrm{TS^{min}_{node}}(P6) = \min(l_\mathrm{u}, |l_\mathrm{d}|) \cdot Q_\mathrm{w} + \mathrm{TS^{min}_{node}}(P1) \tag{25}$$

$$\mathrm{TS^{min}_{device}}(P6) = \mathrm{TS^{min}_{node}}(P6) \tag{26}$$

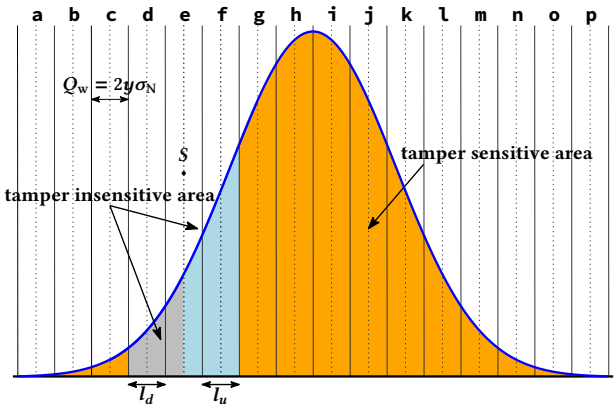


Figure 9: $\mathrm{TS}^{\max}_{\mathrm{node}}$ of Profile 6. Please note the difference to Figure 6 where $\mathrm{TS}^{\min}_{\mathrm{node}}(\mathrm{P4})$ is illustrated, i.e., here we illustrate *max*-TS per node as opposed to *min*-TS in Figure 6.

5 Evaluation of Key Derivation Profiles

In this section we discuss the results listed in Table 1 and Table 2 (cf. Appendix B). All former profiles have been tested based on the empirical data of [TS]. The corresponding parameters are: $\mu_X = 1.8 \cdot 10^{-13}$ and $\sigma_X = 3.6 \cdot 10^{-15}$. Individual measurements of the nodes are affected by Gaussian distributed, mean-free noise with $\sigma_N = 2 \cdot 10^{-16}$.

Starting with Profile 1 in Table 1, it can be seen that even a basic equidistant quantization scheme *without* subsequent ECC is sufficient to create a workable solution. This is achieved by values y of 5.4 or larger, i.e., the width of the quantization intervals needs to be relatively large to account for the assumed noise. We note that the extracted *min*-entropy is only determined by the innermost intervals closest to μ_X, i.e., an increasing number of quantization intervals does *not* increase the *min*-entropy. The extracted entropy ranges in between 267 bits and 231 bits for a reliability in the range of 10^{-6} to 10^{-9}. As described by Equation 4, $\mathrm{TS}^{\max}_{\mathrm{node}}(P1)$ is equivalent to $y \cdot \sigma_N$, whereas $\mathrm{TS}^{\max}_{\mathrm{device}}(P1)$ simply scales this number by the number of nodes v in the system. The corresponding numbers for Profile 1 with the best *max*-TS are therefore 5.4 on a node-level and 692 on a device-level. If we would be considering an increasing number of nodes beyond $v = 128$, it is clear that the increasing numbers of nodes in the exponent of the error probability computation demand an over-excessively wide quantization interval to be counteracted. Hence, this cannot be considered a flexible engineering solution and should only be considered as a baseline for subsequent comparisons. For all subsequent profiles, we investigate whether a smaller y with an additional ECC can perform better than this.

In Profile 2, a fuzzy commitment based on RS codes is used. While y can be lowered to 2.3 resulting in much smaller and more intervals, the helper data leakage caused by the ECC completely counteracts the gain in *min*-entropy such that the effective entropy $\mathrm{H}^{\mathrm{eff}}_{\infty}$ (accounting for the leakage) extracted from the PUF is less than that of Profile 1. In general, this scheme can be adapted easily to different requirements by adjusting t. However, as the distance metric is based on $\mathrm{d}_{\mathrm{H}|S}$, tamper-sensitivity is relatively poor as supported by the obtained results. For both *min*-TS and *max*-TS, the results are actually much worse when compared to a scheme based on equidistant quantization only.

With the help of Profile 3, entropy levels reach a similar amount when compared to Profile 1. This is owed to the differences in the underlying Code-Offset construction when compared to the Fuzzy Commitment scheme, as the leakage is upper bounded by the parity, resulting in a reduced leakage when compared to Profile 2. However, extracting more entropy is at the cost of losing tamper-sensitivity. Moreover, $\mathrm{TS}^{\min}_{\mathrm{node}}$ is *only* defined for

$t = 1$ and therefore represents a strong assumption regarding the attacker as in a practical scenario, the attacker would not be able to divide and distribute the resulting errors to keep them small. Hence, even while the numbers for $\mathrm{TS}^{\min}_{\mathrm{node}}$ indicate a tamper-sensitivity performance close to Profile 1, it cannot be considered a feasible alternative.

For Profile 4, VT-like codes were used with variable-length bit mapping of the symbols. Due to the limitations of these codes, t cannot be chosen arbitrarily and is limited to 1. Consequently, it is not surprising that y cannot be made smaller than 4.24 to still obtain a reliable device. In contrast to Profile 2, a similar *max*-TS is obtained when compared to Profile 1, while performing worse on a node-level. The extracted entropy is marginally better than Profile 1 but we are of the opinion that the added complexity of carrying out the computation for an ECC does not justify this gain.

In contrast to all previous profiles, we applied an equiprobable quantization in Profile 5 which cannot be used as a standalone solution under the given simulation parameters. The given y of 2.87 in the table applies to $Q_{\min}$ only. All other intervals towards $Q_{\max}$ are therefore significantly larger. To provide a fair comparison, we chose to exclude values of X with probability of occurrence less than 0.1%, otherwise, tamper-sensitivity in the outermost intervals would not be bounded which would exaggerate the numbers for *max*-TS unnecessarily. *When neglecting the significant quantization helper data leakage by* W^*, the effective entropy after accounting for the ECC helper data is quite significant, as the equiprobable quantization extracts 3 bits of full entropy per node under this simulated scenario. Regarding tamper-sensitivity, interesting properties are observed. Since the innermost intervals of $Q_{\min}$ are relatively small, the earliest possible detection which translates to *min*-TS on a node-level, is almost within the range of Profile 1. However, most errors that occur are also at or within the range of the innermost intervals. As a result, t must be chosen sufficiently large to account for these errors. This already leads to a suboptimal $\mathrm{TS}^{\min}_{\mathrm{device}}$ behavior. When further analyzing $\mathrm{TS}^{\max}_{\mathrm{node}}$ and $\mathrm{TS}^{\max}_{\mathrm{device}}$, then the obtained tamper-sensitivity performance is clearly worse when compared to Profile 1 and sometimes equally poor when compared to Profile 2 or Profile 4.

Let us now consider our proposal based on equidistant quantization and LMC under the name Profile 6. It can be seen right away that y is the smallest for all considered profiles. For equidistant quantization, this leads to the best-case in terms of entropy that can be extracted from the PUF PDF. Since the equidistant quantization is quite effective in removing a significant portion of the noise influence, only a fraction of nodes need further correction by the LMC. Mainly due to the transformation of q' to p, the overall construction is more efficient when compared to, e.g., Profile 3. This results in a total of ~ 320 effective number of secret bits, the maximum of all previously considered profiles. In addition to that, it can be seen in Table 1 that the per-node *max*-TS is similar to Profile 1 while drastically outperforming all other Profiles. However, the most important result is that *max*-TS on a device level is almost only half of Profile 1. When normalizing $\mathrm{TS}^{\max}_{\mathrm{device}}(\mathrm{P6})$ by the number of nodes, i.e., $395/v = 3.1\,[\sigma_{\mathrm{N}}]$, then this can be interpreted as the on-average tamper detection threshold per node is, $\mathrm{TDT} = 3.1 \cdot \sigma_{\mathrm{N}}$. This is a significant gain in terms of tamper-sensitivity *and* effective number of bits, for various different levels of reliability and alphabet sizes. Taking into account that $\mathrm{TS}^{\min}_{\mathrm{node}}(\mathrm{P6})$ is equal to $\mathrm{TS}^{\max}_{\mathrm{node}}(\mathrm{P6})$ and $\mathrm{TS}^{\min}_{\mathrm{device}}(\mathrm{P6})$ is bounded by the *min*-TS per node of Profile 6, it is evident that the general behavior of LMC mimics the behavior of Profile 1 with regard to the detection of tampering, while performing more effectively which allows to choose a smaller y, resulting in a better entropy and tamper-sensitivity. Overall, this clearly demonstrates the superiority of this scheme and optimized detection of tampering, even when providing a higher reliability when compared to schemes with less reliability.

For the analyzed scenario, in particular the noise model, it appears that this is the optimal method to extract the contained entropy. This is based on the following reasoning: assuming equidistant quantization intervals, then increasing their number towards infinity

causes the entropy to approach the differential entropy which is the entropy defined for the continuous case. Extracting more entropy is not possible. Since LMCs operate more efficiently, quantization intervals can be made smaller and therefore more quantization intervals can be used for the same measurement range. This can be done while the loss in reliability is at a much smaller rate when compared to equiprobable quantization, where the errors increase at a much faster rate when increasing the number of intervals.

6 Conclusion and Outlook

We presented a thorough comparison of state-of-the-art approaches of PUF-based key derivation for tamper-evident PUFs. In addition, we proposed our own scheme that is able to outperform all previous schemes in all relevant aspects, namely: effective number of secret bits, reliability, and tamper-sensitivity. The latter is a new formalized metric to assess the scheme outside of an idealized noise error model based on the noise standard deviation σ_{N}. Intuitively, similar to with physical layer security, sensitivity of a system is bounded by its noise which is why we express this sensitivity in multiples of σ_{N}. We note that the actual assessment of a tamper-evident PUF must still be done individually and that the developed notion of tamper-sensitivity only supports the comparison of the underlying concepts for key generation.

Our comparison clearly demonstrates that the design of previous schemes did not sufficiently take into account the physical tampering that is most important within the context of tamper-evident PUFs. In order to motivate our approach, we provided strong and convincing reasoning as to why the previous schemes fall short in that regard. To overcome their limitations, we applied Limited-Magnitude-Codes based on the Lee/Manhattan metric and a q-ary non-wrap-around channel. Our results indicate that this metric is well-suited to solve the problems specific to tamper-evident PUFs and may generally be able to provide better results when compared to other distance metrics such as Hamming or Levenshtein due to inherent limitations when the number of symbols is less than the number of quantization intervals. We point out that achieving a high level of entropy is still important to thwart attackers that attempt to reconstruct the PUF secret, e.g., by partially obtaining some of its values by probing.

Since our scheme is not limited to custom-made tamper-evident PUFs, we expect that our results can be applied to any other PUF design that allows accessing the quasi-continuous values of the underlying PDF, e.g., [WHGS] or the ones typically implemented in FPGAs such as [BNTM]. Hence, our work may serve as a future direction to improve the area efficiency of these PUFs by reducing the number of PUF primitives and extracting their entropy more efficiently.

Acknowledgements

This work was supported by the Fraunhofer Internal Programs under Grant No. MAVO 828 432. The authors would like to thank Antonia Wachter-Zeh for useful suggestions and feedback on an early version of this article. In addition, we thank our anonymous reviewers and shepherds for improving the quality of the paper.

Table 1: Comparison of key derivation schemes for higher-order alphabet PUFs. Profile settings are shared among publications [IHL+, IHKS, TS] and as follows: $\mu_X = 1.8 \cdot 10^{-13}$ and $\sigma_X = 3.6 \cdot 10^{-15}$. Individual measurements of the nodes are affected by Gaussian distributed, mean-free noise with $\sigma_\mathrm{N} = 2 \cdot 10^{-16}$.

Profile[a]	y	L	z	ECC(n,t)	$\mathbf{P_e}(Y)$ (before ECC)	$\mathbf{P_e}(Y^v)$ (before ECC)	$\mathbf{P_e}(Z^v)$ (after ECC)	$\mathbf{H_\infty^{eff}}$ [bit]	$\mathbf{TS_{node}^{max}}$ [σ_N]	$\mathbf{TS_{device}^{max}}$ [σ_N]	Distance Metric
P1[b]	5.4	8	128	–	6.7×10^{-8}	8.5×10^{-6}	(*id.*)	267	5.4	692	none
	6.6	16	128	–	4.1×10^{-11}	5.3×10^{-9}	(*id.*)	231	6.6	845	
P2[c]	2.3	32	4	RS$(31,7)$	1.2×10^{-2}	7.9×10^{-1}	6.1×10^{-8}	122	148	4352	$\mathrm{d}_{\mathrm{H}\mid S}$
	3	32	4	RS$(31,4)$	2.7×10^{-3}	2.9×10^{-1}	3.4×10^{-7}	193	192	3408	
	5	16	8	RS$(15,1)$	5.7×10^{-7}	7.3×10^{-5}	4.8×10^{-10}	185	160	1880	
P3[d]	2.3	32	4	BCH$(255,8)$	2.1×10^{-2}	9.4×10^{-1}	8.9×10^{-6}	166	148	4932	$\mathrm{d}_{\mathrm{H}\mid 2}$
	2.7	32	7	BCH$(127,4)$	6.9×10^{-3}	5.9×10^{-1}	1.1×10^{-6}	197	173	5109	
	3.6	16	5	BCH$(127,2)$	3.1×10^{-4}	4.0×10^{-2}	1.7×10^{-7}	265	116	1577	
P4[e]	4.95	12	1	VT$(\cdot,1)$	7.4×10^{-7}	9.5×10^{-5}	4.5×10^{-9}	276	65	693	$\mathrm{d}_{\mathrm{Lev}}$
	4.24	14	4	VT$(\cdot,1)$	2.2×10^{-5}	2.8×10^{-3}	1.0×10^{-6}	271	90	828	
P5[f]	2.87	8	2	BCH$(255,7)$	1.3×10^{-3}	1.6×10^{-1}	1.2×10^{-12}	272	112	3558	$\mathrm{d}_{\mathrm{H}\mid 2}$
			2	BCH$(255,4)$			2.8×10^{-7}	320		2994	
P6[g]	2.1	64	1	LMC$(63,10)$	3.6×10^{-2}	9.9×10^{-1}	9.1×10^{-6}	319	6.3	395	$\mathrm{d}_{\mathrm{Man}}$
	2.3	32	1	LMC$(63,9)$	2.1×10^{-2}	9.4×10^{-1}	3.3×10^{-6}	314	6.9	419	
	2.7	32	1	LMC$(63,10)$	6.9×10^{-3}	5.9×10^{-1}	3.7×10^{-12}	273	8.1	508	
	2.7	16	1	LMC$(63,6)$	6.9×10^{-3}	5.9×10^{-1}	3.5×10^{-6}	321	8.1	443	

[a] Neglecting leakage from quantization helper data for computation of $\mathrm{H}_\infty^{\mathrm{eff}}$, i.e., only leakage by ECC helper data W is considered.
[b] **Profile 1 (P1):** Equidistant quantization *without* ECC (independent of symbol's bit mapping)
[c] **Profile 2 (P2):** Equidistant quantization and RS-based Fuzzy Commitment scheme (independent of symbol's bit mapping, n in symbols, t in $\mathrm{d}_{\mathrm{H}\mid S}$)
[d] **Profile 3 (P3):** Equidistant quantization and BCH-based Code-Offset scheme (n in bits, t in $\mathrm{d}_{\mathrm{H}\mid 2}$)
[e] **Profile 4 (P4):** Equidistant quantization, variable-length bit mapping of symbols, VT-like codes (t in $\mathrm{d}_{\mathrm{Lev}}$)
[f] **Profile 5 (P5):** Equiprobable quantization, Gray code bit mapping of symbols, BCH-based Code-Offset scheme (n in bits, t in $\mathrm{d}_{\mathrm{H}\mid 2}$)
[g] **Profile 6 (P6):** Equidistant quantization, LMC ($l_\mathrm{u} = 1, l_\mathrm{d} = -1$) with concatenated RS code (n in symbols, t in $\mathrm{d}_{\mathrm{Man}}$)

References

[AMS+] Frederik Armknecht, Roel Maes, Ahmad-Reza Sadeghi, Francois-Xavier Standaert, and Christian Wachsmann. A formalization of the security features of physical functions. In *IEEE Symposium on Security and Privacy (S&P)*, pages 397–412.

[BGS+] Christoph Bösch, Jorge Guajardo, Ahmad-Reza Sadeghi, Jamshid Shokrollahi, and Pim Tuyls. Efficient helper data key extractor on FPGAs. In Elisabeth Oswald and Pankaj Rohatgi, editors, *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 5154 of *LNCS*, pages 181–197. Springer Berlin / Heidelberg.

[BNTM] M. Barbareschi, G. Di Natale, L. Torres, and A. Mazzeo. A Ring Oscillator-Based Identification Mechanism Immune to Aging and External Working Conditions. 65(2):700–711.

[CBFH] B. Colombier, L. Bossuet, V. Fischer, and D. Hély. Key Reconciliation Protocols for Error Correction of Silicon PUF Responses. 12(8):1988–2002.

[CyCW] J. Chung-yaw Chiang and Jack K. Wolf. On channels and codes for the Lee metric. 19(2):159–173.

[DGV+] Jeroen Delvaux, Dawu Gu, Ingrid Verbauwhede, Matthias Hiller, and Meng-Day (Mandel) Yu. Efficient Fuzzy Extraction of PUF-Induced Secrets: Theory and Applications. In Benedikt Gierlichs and Axel Y. Poschmann, editors, *Cryptographic Hardware and Embedded Systems - CHES 2016 - 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings*, volume 9813 of *Lecture Notes in Computer Science*, pages 412–431. Springer.

[DRS] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Christian Cachin and Jan L. Camenisch, editors, *Advances in Cryptology (EUROCRYPT)*, volume 3027 of *LNCS*, pages 523–540. Springer Berlin / Heidelberg.

[DV] Jeroen Delvaux and Ingrid Verbauwhede. Key-recovery Attacks on Various RO PUF Constructions via Helper Data Manipulation. In *Proceedings of the Conference on Design, Automation & Test in Europe*, DATE '14, pages 72:1–72:6. European Design and Automation Association.

[EB] N. Elarief and B. Bose. Optimal, Systematic,q-Ary Codes Correcting All Asymmetric and Symmetric Errors of Limited Magnitude. 56(3):979–983.

[EFK+] Thomas Esbach, Walter Fumy, Olga Kulikovska, Dominik Merli, Dieter Schuster, and Frederic Stumpf. A New Security Architecture for Smartcards Utilizing PUFs. In *ISSE Conference*.

[GKST] Jorge Guajardo, Sandeep S. Kumar, Geert-Jan Schrijen, and Pim Tuyls. FPGA Intrinsic PUFs and Their Use for IP Protection. In *Cryptographic Hardware and Embedded Systems - CHES 2007*, Lecture Notes in Computer Science, pages 63–80. Springer, Berlin, Heidelberg.

[G O. Günlü and O. İşcan. DCT based ring oscillator Physical Unclonable Functions. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8198–8201.

[HBF] Daniel E. Holcomb, Wayne P. Burleson, and Kevin Fu. Initial SRAM State as a Fingerprint and Source of True Random Numbers for RFID Tags.

[HBNS] C. Helfmeier, C. Boit, D. Nedospasov, and J. Seifert. Cloning Physically Unclonable Functions. In *2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, pages 1–6.

[HNT+] Clemens Helfmeier, Dmitry Nedospasov, Christopher Tarnovsky, Jan Starbug Krissler, Christian Boit, and Jean-Pierre Seifert. Breaking and Entering Through the Silicon. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, CCS '13, pages 733–744. ACM.

[HPKS] Matthias Hiller, Michael Pehl, Gerhard Kramer, and Georg Sigl. Algebraic Security Analysis of Key Generation with Physical Unclonable Functions. https://eprint.iacr.org/2016/854.

[HYP] Matthias Hiller, Meng-Day (Mandel) Yu, and Michael Pehl. Systematic Low Leakage Coding for Physical Unclonable Functions. In Feng Bao, Steven Miller, Jianying Zhou, and Gail-Joon Ahn, editors, *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security, ASIA CCS '15, Singapore, April 14-17, 2015*, pages 155–166. ACM.

[H Matthias Hiller and Aysun Gurur Önalan. Hiding Secrecy Leakage in Leaky Helper Data. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, volume 10529 of *Lecture Notes in Computer Science*, pages 601–619. Springer.

[IHKS] Vincent Immler, Maxim Hennig, Ludwig Kürzinger, and Georg Sigl. Practical Aspects of Quantization and Tamper-Sensitivity for Physically Obfuscated Keys. In *Proceedings of the Third Workshop on Cryptography and Security in Computing Systems*, CS2 '16, pages 13–18. ACM.

[IHL+] Vincent Immler, Matthias Hiller, Qinzhi Liu, Andreas Lenz, and Antonia Wachter-Zeh. Variable-Length Bit Mapping and Error-Correcting Codes for Higher-Order Alphabet PUFs. In *Security, Privacy, and Applied Cryptography Engineering (SPACE)*.

[IOK+18] Vincent Immler, Johannes Obermaier, Martin König, Matthias Hiller, and Georg Sig. B-TREPID: Batteryless tamper-resistant envelope with a PUF and integrity detection. In *IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pages 49–56, 2018.

[ION+] Vincent Immler, Johannes Obermaier, Kuan Kuan Ng, Fei Xiang Ke, JinYu Lee, Yak Peng Lim, Wei Koon Oh, Keng Hoong Wee, and Georg Sigl. Secure Physical Enclosures from Covers with Tamper-Resistance. pages 51–96.

[IW] T. Ignatenko and F. M. J. Willems. Information Leakage in Fuzzy Commitment Schemes. 5(2):337–348.

[JL] M. Jeon and J. Lee. On codes correcting bidirectional limited-magnitude errors for flash memories. In *2012 International Symposium on Information Theory and Its Applications*, pages 96–100.

[JW] Ari Juels and Martin Wattenberg. A Fuzzy Commitment Scheme. In *Proceedings of the 6th ACM Conference on Computer and Communications Security*, CCS '99, pages 28–36. ACM.

[KA] H. Kreft and W. Adi. Cocoon-PUF, a novel mechatronic secure element technology. In *2012 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, pages 227–232.

[LTBS] Heiko Lohrke, Shahin Tajik, Christian Boit, and Jean-Pierre Seifert. No Place to Hide: Contactless Probing of Secret Data on FPGAs. In *Cryptographic Hardware and Embedded Systems – CHES 2016*, Lecture Notes in Computer Science, pages 147–167. Springer, Berlin, Heidelberg.

[Mae] Roel Maes. Physically Unclonable Functions: Constructions, Properties and Applications.

[MCMS] A. Maiti, J. Casarona, L. McHale, and P. Schaumont. A large scale characterization of RO-PUF. In *2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, pages 94–99.

[MGS] Abhranil Maiti, Vikash Gunreddy, and Patrick Schaumont. A Systematic Method to Evaluate and Compare the Performance of Physical Unclonable Functions.

[MTV] Roel Maes, Pim Tuyls, and Ingrid Verbauwhede. Intrinsic PUFs from Flip-flops on Reconfigurable Devices.

[MvdLvdSW] Roel Maes, Vincent van der Leest, Erik van der Sluis, and Frans Willems. Secure Key Generation from Biased PUFs. In *Cryptographic Hardware and Embedded Systems – CHES 2015*, Lecture Notes in Computer Science, pages 517–534. Springer, Berlin, Heidelberg.

[Nat] National Institute of Standards and Technology (NIST). *FIPS PUB 140-2: Security Requirements for Cryptographic Modules*. NIST.

[NSHB] D. Nedospasov, J. Seifert, C. Helfmeier, and C. Boit. Invasive PUF Analysis. In *2013 Workshop on Fault Diagnosis and Tolerance in Cryptography*, pages 30–38.

[OI] Johannes Obermaier and Vincent Immler. The Past, Present, and Future of Physical Security Enclosures: From Battery-Backed Monitoring to PUF-Based Inherent Security and Beyond. pages 1–8.

[SAS] Taras Stanko, Fitria Nur Andini, and Boris Skoric. Optimized Quantization in Zero Leakage Helper Data Systems.

[SD] G. E. Suh and S. Devadas. Physical Unclonable Functions for Device Authentication and Secret Key Generation. In *2007 44th ACM/IEEE Design Automation Conference*, pages 9–14.

[SFIC] M. Spain, B. Fuller, K. Ingols, and R. Cunningham. Robust keys from physical unclonable functions. In *2014 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, pages 88–92.

[SUHA] Manami Suzuki, Rei Ueno, Naofumi Homma, and Takafumi Aoki. Multiple-Valued Debiasing for Physically Unclonable Functions and Its Application to Fuzzy Extractors. In Sylvain Guilley, editor, *Constructive Side-Channel Analysis and Secure Design*, Lecture Notes in Computer Science, pages 248–263. Springer International Publishing.

[TDF+] Shahin Tajik, Enrico Dietz, Sven Frohmann, Helmar Dittrich, Dmitry Nedospasov, Clemens Helfmeier, Jean-Pierre Seifert, Christian Boit, and Heinz-Wilhelm Hübers. Photonic Side-Channel Analysis of Arbiter PUFs. 30(2):550–571.

[Ten] G. Tenengolts. Nonbinary codes, correcting single deletion or insertion (Corresp.). 30(5):766–769.

[TS] Pim Tuyls, Geert-Jan Schrijen, Boris Škorić, Jan van Geloven, Nynke Verhaegh, and Rob Wolters. Read-Proof Hardware from Protective Coatings. In *Cryptographic Hardware and Embedded Systems - CHES 2006*, Lecture Notes in Computer Science, pages 369–383. Springer, Berlin, Heidelberg.

[vN] John von Neumann. Various Techniques Used in Connection With Random Digits.

[VNK+] M. Vai, B. Nahill, J. Kramer, M. Geis, D. Utin, D. Whelihan, and R. Khazan. Secure architecture for embedded systems. In *2015 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–5.

[VT] R. R. Varshamov and G. M. Tenengolts. Codes which correct single asymmetric errors (in Russian). 161(3):288–292.

[VTO+] E. A. Verbitskiy, P. Tuyls, C. Obi, B. Schoenmakers, and B. Skoric. Key Extraction From General Nondiscrete Signals. 5(2):269–279.

[WHGS] Oliver Willers, Christopher Huth, Jorge Guajardo, and Helmut Seidel. MEMS Gyroscopes As Physical Unclonable Functions. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 591–602. ACM.

[YD] Mandel Yu and Srinivas Devadas. Secure and Robust Error Correction for Physical Unclonable Functions. 27(1):48–65.

[YHD] M. Yu, M. Hiller, and S. Devadas. Maximum-likelihood decoding of device-specific multi-bit symbols for reliable key generation. In *2015 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pages 38–43.

[ZHS] Christian Zenger, David Holin, and Lars Steinschulte. Enclosure PUF – Tamper Proofing Commodity Hardware and other Applications. In *35th Chaos Communication Congress (35c3)*.

A Codebooks of Profiles

Since some of the *min*-TS and *max*-TS results depend on the specific codebook chosen to carry out the mapping of symbols to bits, we provide this necessary information to replicate our results. Unfortunately, there is no systematic way provided in [IHL+] on how to construct the codebook for VT-like codes that are relevant for Profile 4. Therefore, the results might be slightly different when choosing a different codebook, while their general behavior will remain the same. All codebooks represent the assigned values to the quantization intervals from "left to right" (cf. Figure 2).

Codebook of Profile 4 [IHL+] and $|\mathcal{L}| = 12$

$\mathcal{L} = [0110; 0111; 0011; 0010; 000; 010; 110; 111; 1011; 1010; 1000; 1001]$
The maximum magnitude shift while ensuring $\mathrm{d}_{\mathrm{Lev}}(Y, \hat{Y}) = 1$ may occur for the following values, all of which describe a shift by 6 quantization intervals:

- $0110 \leftrightarrow 110$ (insertion/deletion of 0)
- $0111 \leftrightarrow 111$ (insertion/deletion of 0)
- $000 \leftrightarrow 1000$ (insertion/deletion of 1)
- $0011 \leftrightarrow 1011$ (substitution of 0/1 in left most position)
- $0010 \leftrightarrow 1010$ (substitution of 0/1 in left most position)

Codebook of Profile 4 [IHL+] and $|\mathcal{L}| = 14$

$\mathcal{L} = [01100; 01101; 0111; 0011; 0010; 000; 010; 110; 111; 1011; 1010; 1000; 10010; 10011]$
The maximum magnitude shift while ensuring $\mathrm{d}_{\mathrm{Lev}}(Y, \hat{Y}) = 1$ may occur for the following values which describe a shift by 10 quantization intervals:

- $0011 \leftrightarrow 10011$ (insertion/deletion of 1)

Codebook of Profile 5 [TS] and $|\mathcal{L}| = 8$

$\mathcal{L} = [000; 001; 011; 010; 110; 111; 101; 100]$
Larger magnitude shifts exceeding the range of one quantization interval while still ensuring $\mathrm{d}_{\mathrm{H}|2}(Y, \hat{Y}) = 1$ may occur for the following values:

- $000 \leftrightarrow 010$ (shift by 3 quantization intervals of unequal size)
- $011 \leftrightarrow 111$ (shift by 3 quantization intervals of unequal size)
- $110 \leftrightarrow 100$ (shift by 3 quantization intervals of unequal size)
- $001 \leftrightarrow 101$ (shift by 5 quantization intervals of unequal size)
- $000 \leftrightarrow 100$ (shift is across the full range of values)

B Additional Tamper-Sensitivity Results

The following table complements the tamper-sensitivity results of Table 1 regarding *min*-TS and also provides the numbers for *max*-TS normalized by the number of nodes v (last column), therefore representing the on-average per-node sensitivity. These numbers enable a comparison across different tamper-evident PUF system designs with varying number of PUF nodes v.

Table 2: Extended comparison of TS of different profiles for higher-order alphabet PUFs, including *min*-TS results and *max*-TS results normalized by the number of nodes $v = 128$ in the tamper-evident PUF system.

Profile	y	L	z	ECC(n,t)	$\mathrm{P_e}(\mathbf{Z}^\mathbf{v})$	$\mathrm{H}_\infty^{\mathrm{eff}}$ [bit]	$\mathrm{TS}_{\mathrm{node}}^{\mathrm{min}}$ [σ_N]	$\mathrm{TS}_{\mathrm{node}}^{\mathrm{max}}$ [σ_N]	$\mathrm{TS}_{\mathrm{device}}^{\mathrm{min}}$ [σ_N]	$\mathrm{TS}_{\mathrm{device}}^{\mathrm{max}}$ [σ_N]	$\mathrm{TS}_{\mathrm{device}}^{\mathrm{max}}/\mathrm{v}$ [σ_N]
P1	5.4	8	128	–	8.5×10^{-6}	267	5.4	5.4	5.4	692	5.4
	6.6	16		–	5.3×10^{-9}	231	6.6	6.6	6.6	845	6.6
P2	2.3	32	4	RS$(31,7)$	6.1×10^{-8}	122	∞	148	4124	4352	34
	3	32	4	RS$(31,4)$	3.4×10^{-7}	193	∞	192	3075	3408	27
	5	16	8	RS$(15,1)$	4.8×10^{-10}	185	∞	160	1285	1880	15
P3	2.3	32	4	BCH$(255,8)$	8.9×10^{-6}	166	6.9	148	224	4932	39
	2.7	32	7	BCH$(127,4)$	1.1×10^{-6}	197	8.1	173	230	5109	40
	3.6	16	5	BCH$(127,2)$	1.7×10^{-7}	265	10.8	116	112	1577	13
P4	4.95	12	1	VT$(\cdot,1)$	4.5×10^{-9}	276	15	65	15	693	5.4
	4.24	14	4	VT$(\cdot,1)$	1.0×10^{-6}	271	13	90	13	882	6.9
P5	2.87	8	2	BCH$(255,7)$	1.2×10^{-12}	272	8.7	112	141	3558	30
				BCH$(255,5)$	2.8×10^{-7}	320			72	2994	24
P6	2.1	64	1	LMC$(63,10)$	9.1×10^{-6}	319	6.3	6.3	6.3	395	3.1
	2.3	32	1	LMC$(63,9)$	3.3×10^{-6}	314	6.9	6.9	6.9	419	3.3
	2.7	32	1	LMC$(63,10)$	3.7×10^{-12}	273	8.1	8.1	8.1	508	4.0
	2.7	16	1	LMC$(63,6)$	3.5×10^{-6}	321	8.1	8.1	8.1	443	3.5

C PUF Metrics based on Lee/Manhattan Metric

Originally defined by Maiti et al. in [MGS, MCMS], the metrics *Uniqueness* and *Reliability* based on Hamming distance have become the de facto standard for PUF publications. While various other metrics have been proposed, they can still be considered as a starting point to assess the fundamental PUF properties, i.e., if PUF values sufficiently differ from each other and if they can be reconstructed reliably. While we recommend to always complement them with additional tests, we nevertheless focus on these two most common metrics with regard to higher-order alphabets which is owed to their popularity.

The definition of Uniqueness and Reliability is inherently bound to the distance metric used for the subsequent ECC. For higher-order alphabet based PUFs over Hamming distance this has been studied in [ION+] already. In contrast, we define Uniqueness and Reliability over the Lee or Manhattan distance, depending on the q-ary channel model used, i.e., whether it is used with wrap-around or without.

We note that Uniqueness and Reliability in the binary case, as originally defined by [MGS, MCMS], is normalized by the length v of the considered response (cf. Figure 1). This must be done in an appropriate manner also for responses over Lee/Manhattan distance, as their length is different due to how the distance metrics $\mathrm{d}_{\mathrm{Lee}}$ and $\mathrm{d}_{\mathrm{Man}}$ are defined. Lee distance $\mathrm{d}_{\mathrm{Lee}}$ between two quantized PUF responses, with a field size of q, is defined below in Equation (27). It is circular i.e., $\mathrm{d}_{\mathrm{Lee}}(0, q-1) = 1$.

$$\mathrm{d}_{\mathrm{Lee}}(Y_1^v, Y_2^v) = \sum_{i=1}^{v} \min((y_i^1 - y_i^2), q - (y_i^1 - y_i^2)) \tag{27}$$

Similar to before, Manhattan distance $\mathrm{d}_{\mathrm{Man}}$ between two words is defined below in Equation (28). It is non-circular, i.e., $\mathrm{d}_{\mathrm{Lee}}(0, q-1) = q-1$.

$$\mathrm{d}_{\mathrm{Man}}(Y_1^v, Y_2^v) = \sum_{i=1}^{v} |y_i^1 - y_i^2| \tag{28}$$

where $Y_j^v = \{y_i^j; 1 \le i \le v\}$, $j = 1, 2$ and $0 \le y_i^j \le q-1$ For LMCs in order to normalize, we apply Plotkin's low rate average distance bound defined in Equation (29) for the wrap-around channel [CyCW].

$$\mathrm{d}_{\mathrm{Lee}} \le \frac{v\overline{D}}{(1 - K^{-1})} \tag{29}$$

where K is the cardinality of $\mathcal{C}$ and $\overline{D}$ is the average Lee weight [CyCW] given by Equation (30).

$$\overline{D} = \begin{cases} \frac{(q^2-1)}{4q}, & \text{odd q} \\ \frac{q}{4}, & \text{even q} \end{cases} \tag{30}$$

For the practical scenario [ION+] of $v = 128$ nodes in a PUF device with field size $q = 32$, $K = q^{128}$ this leads to $K^{-1} \approx 0$. Thus Equation (31) holds which makes it compatible to previous definitions of Uniqueness for binary PUFs [MGS].

$$\frac{\mathrm{d}_{\mathrm{Lee}}}{v\overline{D}} \le 1 \tag{31}$$

For Uniqueness over Manhattan distance, the same is achieved by normalizing the length with $v \cdot q$. To define the Uniqueness, this leads to Equation (32) for Lee distance and Equation (33) for Manhattan distance.

$$\text{Uniqueness}_{\text{d}_{\text{Lee}}} = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \frac{\text{d}_{\text{Lee}}(Y_i^v, Y_j^v)}{v\ \overline{D}} \times 100\% \tag{32}$$

$$\text{Uniqueness}_{\text{d}_{\text{Man}}} = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \frac{\text{d}_{\text{Man}}(Y_i^v, Y_j^v)}{v\ q} \times 100\% \tag{33}$$

where k is the number of devices and v the number of nodes in a device which is equivalent to its length in symbols. Please note that the computed outcome of these definitions is quite different to the ones based on Hamming Distance, as the magnitude becomes part of the Uniqueness which is no longer just a change in symbol. Hence, to improve Uniqueness, not only the symbols as such would have to change, but also the occurred magnitude. Additionally note that the computed outcome of Equation (32) compared to Equation (33) is quite different, since the normalization factor in front of the sum remains unchanged.

In particular, the result of Equation (33) is such that the Uniqueness is bounded to 100% which would be achieved only when for each symbol a maximum magnitude change is observed. In contrast, empirical results for Equation (32) may exceed 100% of Uniqueness when the average Lee weight of Equation (30) is exceeded.

To complement the previous Uniqueness definitions, Reliability is defined for both metrics in Equation (34) and Equation (35)

$$\text{Reliability}_{\text{d}_{\text{Lee}}} = \frac{1}{m} \sum_{i=1}^{m} \frac{\text{d}_{\text{Lee}}(Y_i^v, Y_{i,t}^v)}{v\ \overline{D}} \times 100\% \tag{34}$$

$$\text{Reliability}_{\text{d}_{\text{Man}}} = \frac{1}{m} \sum_{i=1}^{m} \frac{\text{d}_{\text{Man}}(Y_i^v, Y_{i,t}^v)}{v\ q} \times 100\% \tag{35}$$

where m is the number of measurements of same PUF device at different times. Since the channel model corresponding to the work in [ION+] is without wrap-around, we select Manhattan Distance as the appropriate distance metric. This leads to the results presented in Figure 10, showing both Uniqueness and Reliability. In contrast to similar figures for binary PUFs, Uniqueness appears relatively low which is owed to the fundamentally different definition of Uniqueness over Manhattan distance that combines changes in symbols and magnitude at the same time. To put the outcome into perspective note that for the given parameters, a change in 3.125% corresponds to the case when all comparisons between symbols result in a magnitude of $\text{d}_{\text{Man}} = 1$. For the given data, the average Uniqueness is 21.897% which is very close to the case that every compared symbol has a distance $\text{d}_{\text{Man}} = 7$ which corresponds to 21.875% of Uniqueness. Considering the fact that this is the first such implementation which differs quite significantly from other PUFs, Uniqueness appears at a reasonable level which could be improved though to make it more unique. In contrast, Reliability is at a very high level.

D Example Calculations of LMC

For convenience reasons and clarity of the algorithmic descriptions, we provide examples of several LMC calculations for the interested reader. They follow the notation of Algorithm 1 and Algorithm 2. Please note that these calculations are based on $q' = 4$, i.e., the radix is a power of two, allowing for a very efficient hardware implementation. These examples include:

- Figure 11 illustrating the enrollment process.
- Figure 12 illustrating a successful decoding of the previous example.

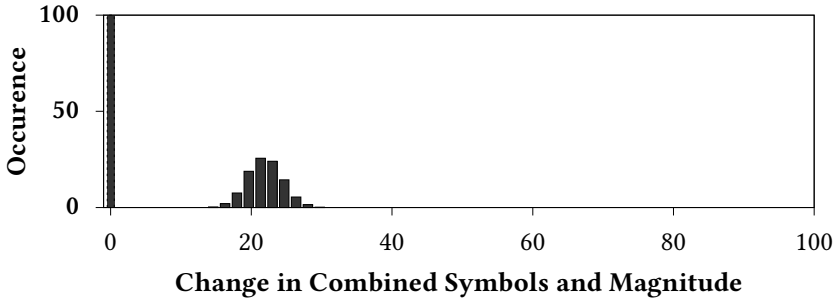


Figure 10: Uniqueness and Reliability according to Equation (33) and Equation (35) based on the evaluation of 115 covers of [ION+] with 32 quantization intervals and a 10× oversampling for the measurement.

- Figure 13 illustrating a decoding failure as result of an error exceeding the magnitude.

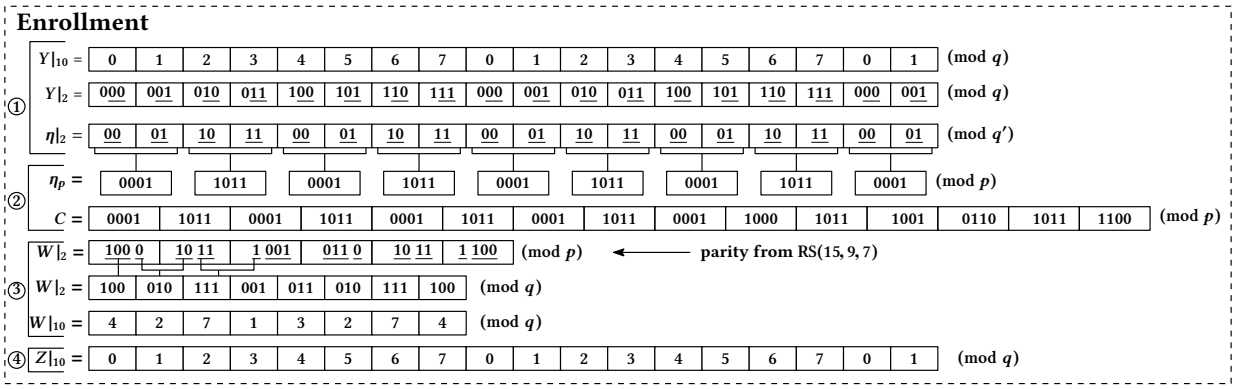


Figure 11: LMC encode example (q=8, $q' = 4$, $l_u = 2, l_d = -1$, p=16).

Figure 12: LMC example for successful decoding (q=8, $q' = 4$, $l_u = 2, l_d = -1$, p=16).

Figure 13: LMC example for decoding failure (q=8, $q' = 4$, $l_u = 2, l_d = -1$, p=16).