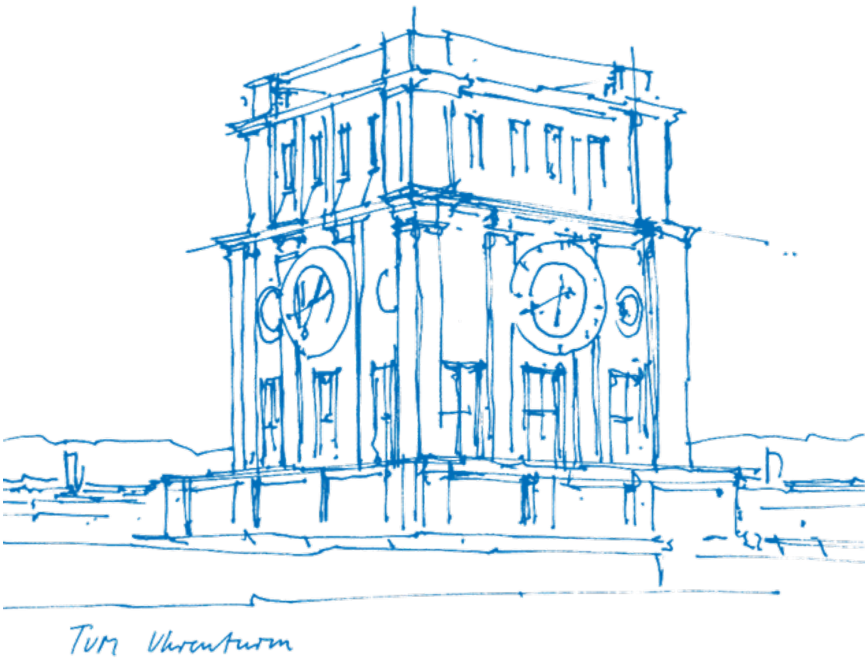
Chair for Security in Information Technology
School of Computation, Information and Technology
Technical University of Munich

Towards Efficient Helper Data Algorithms for Multi-Bit PUF Quantization

Marius Drechsler



Chair for Security in Information Technology
School of Computation, Information and Technology
Technical University of Munich

Towards Efficient Helper Data Algorithms for Multi-Bit PUF Quantization

Marius Drechsler

Thesis for the attainment of the academic degree

Bachelor of Science (B.Sc.)

at the School of Computation, Information and Technology of the Technical University of Munich.

Examiner:
Prof. Dr. Georg Sigl

Supervisor:
M.Sc. Jonas Ruchti

Submitted:
Munich, 22.07.2024

Contents

1 Introduction

These are the introducing words

1.1 Notation

To ensure a consistent notation of functions and ideas, we will now introduce some required conventions

Random distributed variables will be notated with a capital letter, i.e. X, its realization will be the corresponding lower case letter, x.

Vectors will be written in bold test: $\boldsymbol{k}$ represents a vector of quantized symbols.

We will call a quantized symbol k. k consists of all possible binary symbols, i.e. $0, 01, 110$.

A quantizer will be defined as a function $\mathcal{Q}(x, \boldsymbol{a})$ that returns a quantized symbol k. We also define the following special quantizers for metric based HDAs: A quantizer used during the enrollment phase is defined by a calligraphic $\mathcal{E}$. For the reconstruction phase, a quantizer will be defined by a calligraphic $\mathcal{R}$

Figure 1 shows the curve of a 2-bit quantizer that receives $\tilde{x}$ as input. In the case, that the value of $\tilde{x}$ equals one of the four bounds, the quantized value is chosen randomly from the relevant bins.

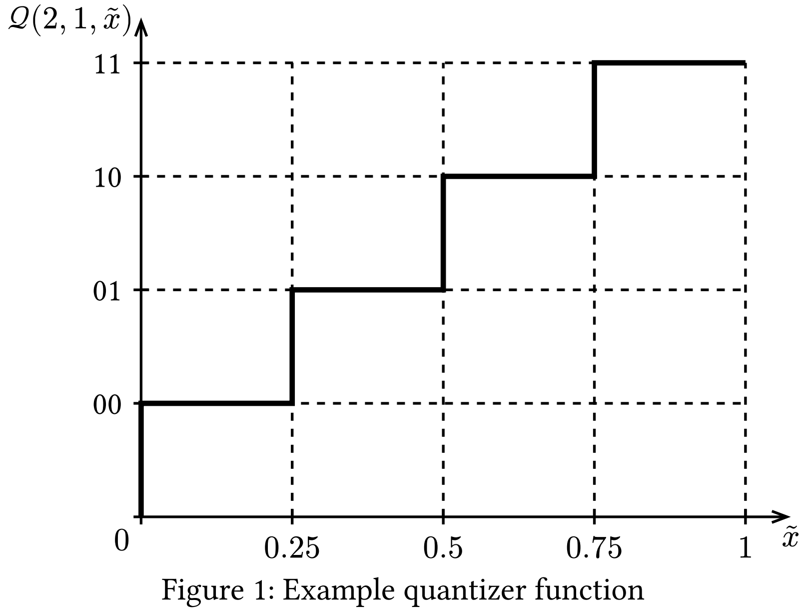


Figure 1: Example quantizer function

For the S-Metric Helper Data Method, we introduce a function

$$\mathcal{Q}(s, m) \tag{1}$$

where s determines the amount of metrics and m the bit width of the symbols.

1.1.1 Tilde-Domain

AS also described in (Fischer), we will use a CDF to transform the real PUF values into the Tilde-Domain This transformation can be performed using the function $\xi = \tilde{x}$. The key property of this transformation is the resulting uniform distribution of x.

Considering a normal distribution, the CDF is defined as

$$\xi\left(\frac{x-\mu}{\sigma}\right) = \frac{1}{2}\left[1 + erf\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right] \tag{2}$$

1.1.2 ECDF

The eCDF is constructed through sorting the empirical measurements of a distribution (Dekking). Although less accurate, this method allows a more simple and less computationally complex way to transform real valued measurements into the Tilde-Domain. We will mainly use the eCDF in Section 2 because of the difficulty of finding an analytical description for the CDF of a Gaussian-Mixture.

2 S-Metric Helper Data Method

A metric based helpder data algorithm (HDA) generates helper data at PUF enrollment to provide more reliable results at the reconstruction stage. Each of these metrics correspond to a quantizer with different bounds to lower the risk of bit or symbol errors during reconstruction.

2.1 Background

2.1.1 Distribution Independency

The publications for the Two-Metric approach (Danger et al.) and (Tebelmann et al.), as well as the generalized S-Metric approach (Fischer) make the assumption, that the PUF readout is "zero-mean Gaussian distributed" (Fischer). We propose, that a Gaussian distributed input für S-Metric quantization is not required for the operation of this quantizing algorithm. Instead, any distribution can be used for input values given, that a CDF exists for that distribution and its parameters are known. As already mentioned in Section 1.1.1, this transformation will result in uniformly distributed values, where equi-probable areas in the real domain correspond to equi-distant areas in the Tilde-Domain. Contrary to (Danger et al.), (Tebelmann et al.) and (Fischer), which display relevant areas as equi-probable in a normal distribution, we will use equi-distant areas in a uniform distribution for better understandability. It has to be mentioned, that instead of transforming all values of the PUF readout into the Tilde-Domain, we could also use an inverse CDF to transform the bounds of our evenly spaced areas into the real domain with (normal) distributed values, which can be assessed as remarkably less computationally complex.

2.1.2 Two-Metric Helper Data Method

The most simple form of a metric-based HDA is the Two-Metric Helper Data Method, since the quantization only yields symbols of 1-bit width and uses the lead amount of metrics possible. Publications (Danger et al.) and (Tebelmann et al.) find all the relevant bounds for the enrollment and reconstruction phases under the assumption that the PUF readout is Gaussian distributed. Because this approach is static, meaning the parameters for symbol width and number of metrics always stays the same, it is easier to calculate the bounds for 8 equi-probable areas with a standard deviation of $\sigma = 1$ first and then multiplying them with the estimated standard deviation of the PUF readout. This is done by fining two bounds a and b, that

$$\int_a^b f_{X(x)} dx = \frac{1}{8} \tag{3}$$

This operation yields 9 bounds defining these areas $-T1$, $-a$, $-T2$, 0, $T2$, a, $T1$ and $\pm\infty$ During the enrollment phase, we will $\pm a$ as our quantizing bounds, retuning 0 if the absolute values is smaller than a and 1 otherwise. The corresponding metric is chosen based on the following conditions:

$$M = \begin{cases} M1, x < -a \vee 0 < x < a \\ M2, -a < x \vee 1 < a < x \end{cases} \tag{4}$$

Figure 2 shows the curve of a quantizer $\mathcal{Q}$, that would be used during the Two-Metric enrollment phase. At this point, we will still assume, that our input value x is zero-mean Gaussian distributed.

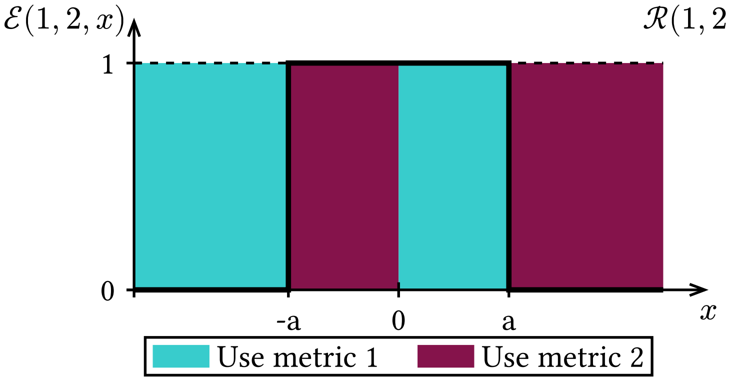


Figure 2: Two-Metric enrollment

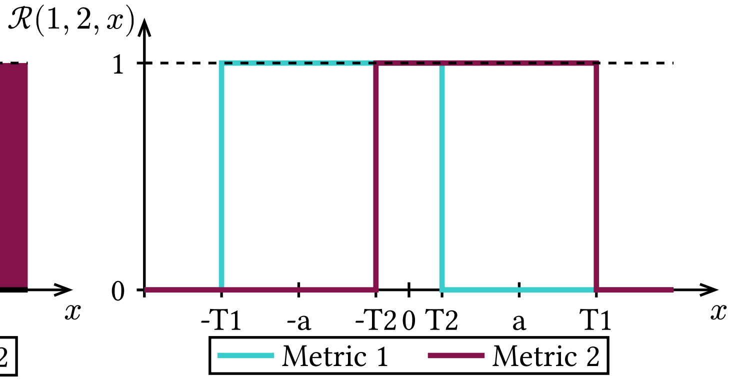


Figure 3: Two-Metric reconstruction

The metric will be stored publicly for every quantized bit as helper data. As previously described, each of these metrics correspond to a different quantizer. Now, we can use the generated helper data in the reconstruction phase and define a reconstructed bit based on the chosen metric as follows:

$$M1 : k = \begin{cases} 0, x < T1 \lor T2 < x \\ 1, -T1 < x < T2 \end{cases} \qquad M2 : k = \begin{cases} 0, x < -T2 \lor T1 < x \\ 1, -T2 < x < T1 \end{cases} \tag{5}$$

Figure 3 illustrates the basic idea behind the Two-Metric method. Using the helper data, we will move the bounds of the original quantizer one octile to each side, yielding two new quantizers. The advantage of this method comes from moving the point of uncertainty away from our readout position.

Figure 4 and Figure 5 illustrate an example enrollment and reconstruction process. We would consider the marked point the value of the initial measurement and the marked range our margin of error due to inaccuracies in the measurement process. If we now were to use the quantizer shown in Figure 4 during both the enrollment and the reconstruction phases, we would risk a bit error, because the margin of error overlaps with the lower quantization bound $-a$. But since we generated helper data during enrollment as depicted in Figure 2, we can make use of a different quantizer $\mathcal{R}(1, 2, x)$ whose boundaries do not overlap with the error margin of the measurement.

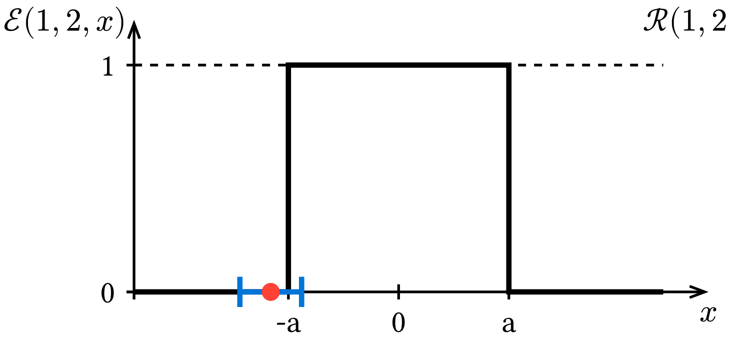


Figure 4: Example enrollment

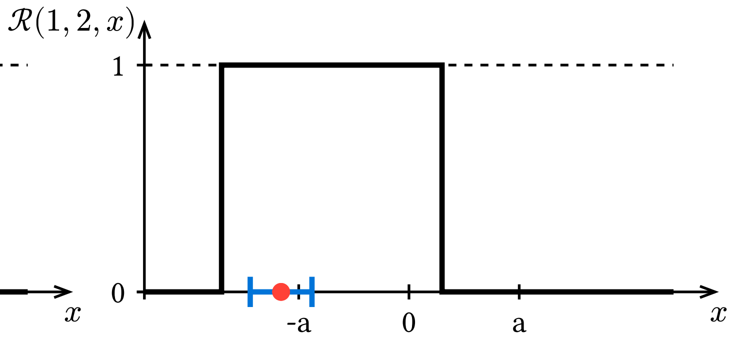


Figure 5: Example reconstruction

2.1.3 S-Metric Helper Data Method

Going on, the Two-Metric Helper Data Method can be generalized as shown in (Fischer). This generalization allows for higher order bit quantization and the use of more than two metrics.

A key difference to the Two-Metric approach is the alignment of quantization areas. Methods described in (Danger et al.) and (Tebelmann et al.) use two bounds for 1-bit quantization, namely $\pm a$. Contrary, the method introduced by (Fischer) would look more like a sign based quantizer if the configuration $\mathcal{Q}(2, 1)$ is used, using only one quantization bound at $x = 0$. Figure 6 and Figure 7 illustrate this difference.

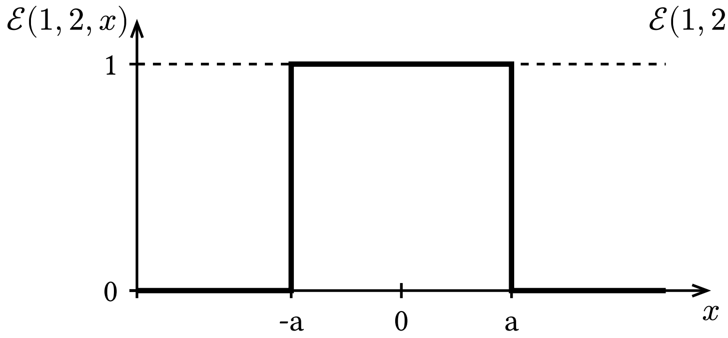


Figure 6: Two-Metric enrollment

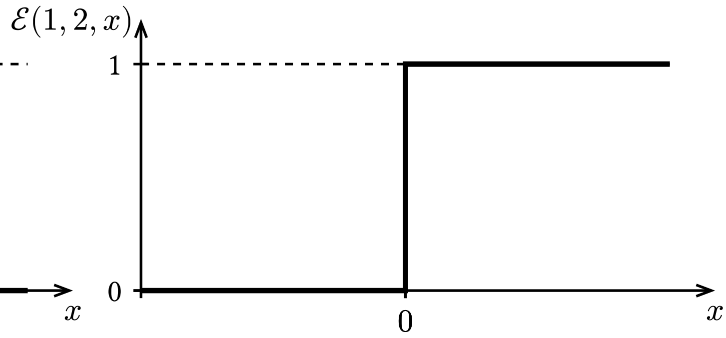


Figure 7: S-Metric enrollment with 1-bit configuration

The generalization consists of two components:

- **Higher order bit quantization**
  We can introduce more steps to our quantizer and use them to extract more than one bit out of our PUF readout.
- **Using more than two metrics**
  Instead of splitting each quantizer steam into only two equi-probable parts, we can increase the number of metrics at the cost of generating more helper data.

2.2 Implementation

We will now propose a specific implementation of the S-Metric Helper Data Method.
As shown in Section 2.1.1, we can use a CDF to transform our random distributed variable X into the Tilde-Domain: $\tilde{X}$. This allows us to use equi-distant bounds for the quantizer instead of equi-probable ones.

From now on we will use the following syntax for quantizers that use the S-Metric Helper Data Method:

$$\mathcal{Q}(s, m, \tilde{x}) \tag{6}$$

where s defines the number of metrics, m the number of bits and $\tilde{x}$ a Tilde-Domain transformed PUF measurement.

2.2.1 Enrollment

To enroll our PUF key, we will first need to define the quantizer for higher order bit quantization and helper data generation. Because our PUF readout $\tilde{x}$ can be interpreted as a realization of a uniformly distributed variable $\tilde{X}$, we can define the width Δ of our quantizer bins as follows:

$$\Delta = \frac{1}{2^m} \tag{7}$$

For example, if we were to extract a symbol with the width of 2 bits from our PUF readout, we would need to evenly space $2^2 = 4$ bins. Using equation Equation 7, the step size for a 2-bit quantizer would result to:

$$\Delta' = \frac{1}{2^m}\bigg|_{m=2} = \frac{1}{4} \tag{8}$$

Figure 8 shows a plot of the resulting quantizer function that would yield symbols with two bits for one measurement $\tilde{x}$.

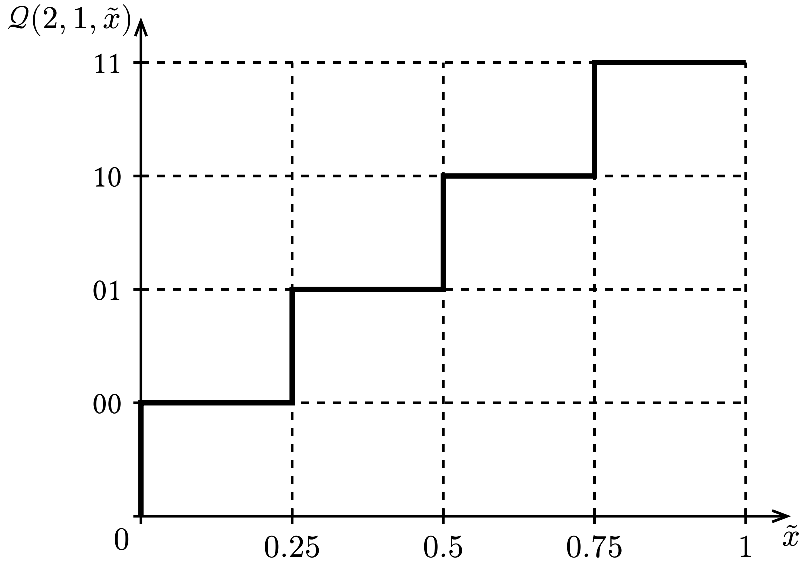


Figure 8: 2-bit quantizer

Right now, this quantizer wouldn't help us generating any helper data. To achieve that, we will need to divide a symbol step - one, that returns the corresponding quantized symbol - into multiple sub-steps. More specifically, we will define the amount of metrics we want to use with the parameter s. Using s, we can define the step size Δ_s as the division of Δ by s:

$$\Delta_s = \frac{\Delta}{s} = \frac{\frac{1}{2^m}}{s} = \frac{1}{2^m * s} \tag{9}$$

After this definition, we need to make an adjustment to our previously defined quantizer function, because we cannot simply return the quantized value based on a quantizer with step size Δ_s. That would just increase the amounts of bits we will extract out of one measurement. Instead, we will need to return a tuple, consisting of the quantized symbol and the metric ascertained that we will save as helper data for later.

Going on in our example, we could choose the amount of our metrics to be 2. According to Equation 9, we would then half out step size:

$$\Delta'_s = \frac{\Delta'}{s}\bigg|_{s=2} = \frac{1}{4*2} = \frac{1}{8} \tag{10}$$

This means, we can update out quantizer function with the new step size $\Delta'_s = \frac{1}{8}$ and redefining its output as a tuple consisting of bit value and helper data.

We can visualize the quantizer that we will use during the enrollment phase of a 2-bit 2-metric configuration as depicted in Figure 9.

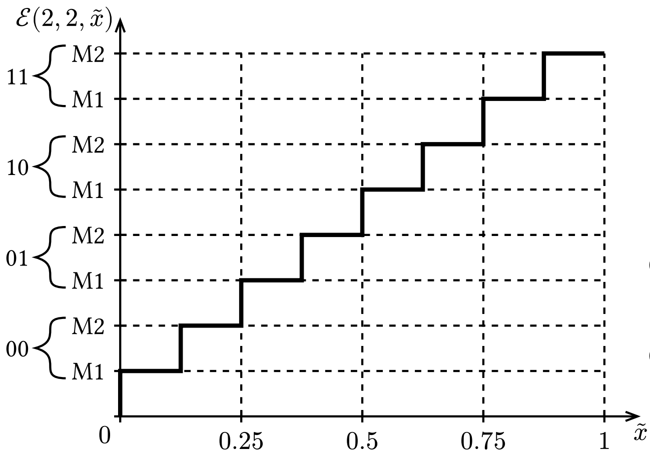


Figure 9: 2-bit 2-metric enrollment

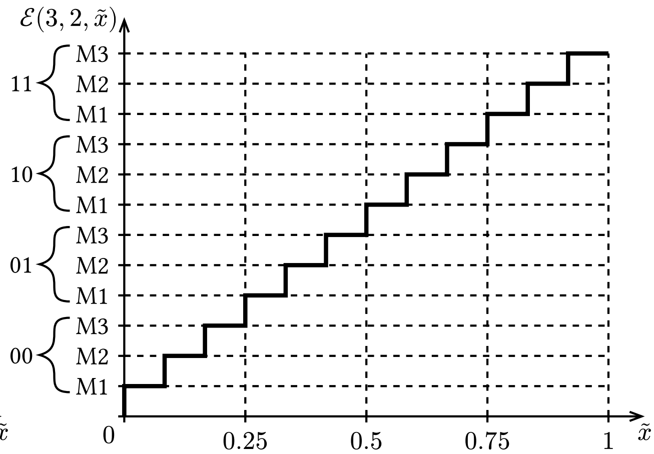


Figure 10: 2-bit 3-metric enrollment

To better demonstrate the generalization to s-metrics, Figure 10 shows a 2-bit quantizer that generates helper data based on three metrics instead of two. In that sense, increasing the number of metrics will increase the number of sub-steps for each symbol.

We can now perform the enrollment of a full PUF readout. Each measurement will be quantized with out quantizer $\mathcal{E}$, returning a tuple consisting of the quantized symbol and helper data, as shown in Equation 11

$$K_i = \mathcal{E}(s, m\tilde{x}_i) = (k, h)_i \tag{11}$$

Performing the operation of Equation 11 for our whole set of measurements will yield a vector of tuples $\boldsymbol{K}$.

Glossary

***HDA* – helpder data algorithm**. 9

Bibliography

Danger, Jean-Luc, et al. "Two-Metric Helper Data for Highly Robust and Secure Delay Pufs." *2019 IEEE 8th International Workshop on Advances in Sensors and Interfaces (IWASI)*, 2019, pp. 184–88

Dekking, Frederik Michel. *A Modern Introduction to Probability and Statistics: Understanding Why and How*. Springer Science & Business Media, 2005

Fischer, Robert FH. "Helper Data Schemes for Coded Modulation and Shaping in Physical Unclonable Functions." *Arxiv Preprint Arxiv:2402.18980*, 2024

Tebelmann, Lars, et al. "Analysis and Protection of the Two-Metric Helper Data Scheme." *International Workshop on Constructive Side-Channel Analysis and Secure Design*, 2021, pp. 279–302