An Efficient Digitizer for Gaussian Physically Unclonable Functions

Riccardo Bernardini[1]

[1]Affiliation not available

March 28, 2025

An Efficient Digitizer for Gaussian Physically Unclonable Functions

Riccardo Bernardini

March 26, 2025

Abstract

Analog Physically Unclonable Functions is a new class of Physically Unclonable Functions with the characteristic that the output is not a bitword (as it is in the traditional Physically Unclonable Functions), but an analog value that is successively converted to digital by a suitable circuit called *digitizer*. The advantage is that, potentially, if a suitable digitizer is employed, a single cell can produce more than one bit, simplfying the implementation and reducing the footprint. This paper proposes a simple digitizer for Gaussian Analog Physically Unclonable Functions that exploits the properties of quantization noise to obtain a stable and uniformly distributed output.

1 Introduction

The problem of providing secure "fingerprinting" to chip (e.g., for authentication purpose) stimulated the development of several Physically Unclonable Function (PUF) schemes [1–11]. A PUF is a circuit that implements a function that maps bit-strings to bit-strings. The peculiarity that characterizes a PUF is its sensitivity to components variations, making the behavior of every PUF unique and akin to a fingerprint of the chip.

A small class of PUF is represented by the PUFs that have no input and, therefore, always return the same value each time they are queried. This kind of PUF is sometimes called Physically Unclonable Constant (PUC) or Physically Obfuscated Key (POK) and it can be interpreted as a way to embed a random secret in a chip. The secret embedded with a PUC can be used in any criptographic protocol: for example, as the private secret in a Diffie-Hellman (DH) scheme, as the private key for asymmetric cryptography, ans so on. We will say that an *ideal* PUC is a *random constant* [12, 13] since a PUC always returns the same value that has been randomly "selected" at construction time.

A PUC is a *two-step experiment*: the first step happens at construction time and randomly select the value that the PUC should return every time it is queried (called *nominal output* in the following); the second step takes place when the PUC generates the output that can differ from the nominal one because of noise. An intuitive model for a two-step experiment is the *bag of coins*: we have a bag that contains many coins, some coin is biased toward head, some other toward tails and maybe few coins are unbiased. At construction time a coin is randomly chosen and every time the PUC is queried the coin is flipped.

Most PUC schemes are digital circuits that when they are turned on, they converge, after a transient, to one of two possible states. PUCs can be based on comparators [14, 15], SRAM cells or latches [16–20, 20–23]. Finally, there are schemes like [24–26] where the PUC is a circuit that, differently from memories that have *two* stable states by design, have a single stable state whose position is very sensitive to the circuit parameters. All these schemes are digital.

Recently Analog-based Physically Unclonable Constants (APUCs) have been proposed [27]. APUCs obtain the secret bit-string by measuring some analog value affected by construction time dispersion, for example, resistance [27]. The advantages of APUCs are the possibility of getting more bit per cell (while digital based PUCs produce usually one bit per cell) and a potentially very stable result, if the measurement is done correctly. A drawback of APUCs is that usually the analog values are not uniformly distributed (most often they are normally distributed); therefore, the bits obtained by the measurement are not unbiased and iid. This issue can be solved by employing a *conditioner*, that is, a processing block that "distillate" the randomness in the measurement to produce Unbiased and Independent Identically Distributed (uiid) bits. Since the outcome of measurements are binary words, one can employ any generic conditioning scheme, for example, Elias-like schemes [28–34] that are complex, but give *strong theoretical guarantees* about the quality of the generated bits or simpler solutions such as hash functions [35] that are simpler, but with less strong theoretical guarantees (often based on assumptions like of being *random oracles*).

In this paper we propose a conditioning scheme suited for APUC that is very simple to implement and gives strong guarantees about the randomness of the result.

2 Preliminaries

2.1 Notation

Translation and scaling In few proofs it will be convenient to use some special notation for the operations of translating and scaling a function $f : \mathbb{R} \to \mathbb{R}$. For technical reasons, we assume f to be absolutely summable, that is $f \in L^1(\mathbb{R})$.

We will denote f translated by $d \in \mathbb{R}$ as $\tau^d f$ and f rescaled by $\lambda \neq 0$ as $S_\lambda f$. More precisely, operators

$\tau^d : L^1(\mathbb{R}) \to L^1(\mathbb{R})$ and $S_\lambda : L^1(\mathbb{R}) \to L^1(\mathbb{R})$ are defined as follows

$$\tau^d f(x) = f(x-d) \tag{1a}$$

$$S_\lambda f(x) = |\lambda|\, f(\lambda x) \tag{1b}$$

Note that with definition (1) $\|f\|_1 = \|\tau^d f\|_1 = \|S_\lambda f\|_1$. This notation will make some derivation easier, especially because of the following properties whose proof is immediate

$$\tau^a \tau^b = \tau^{a+b} \qquad S_\lambda S_\mu = S_{\lambda\mu} \qquad S_\lambda \tau^{\lambda d} = \tau^d S_\lambda \tag{2}$$

Random variables If X is a *random variable*, we will denote its Probability Density Function (PDF) as f_x and its Cumulative Distribution Function (CDF) as F_X. If X has mean m and variance σ^2, we define its *normalized version* as $\overline{X} = (X-m)/\sigma$; in other words, $\overline{X}$ has the same "shape" as X, but it has zero mean and unitary variance. The PDF of $\overline{X}$ is clearly $f_{\overline{X}}(x) = \sigma f_X(\sigma x + m)$, or, using operators (1) $f_{\overline{X}} = S_\sigma \tau^{-m} f_X$.

If X is a Gaussian random variable with mean m and variance σ^2 we will write $X \sim \mathcal{N}(m, \sigma^2)$; if X is random variable uniformly distributed on set $I \subset \mathbb{R}$ we will write $X \sim \mathcal{U}(I)$. Sometimes we will use notations like $\mathcal{N}(m, \sigma^2)$ as an "anonymous" random variable, e.g. $P[\mathcal{N}(m, \sigma^2) > 1/2]$.

In the following we will often assume that a given random variable X satisfies the following hypothesis.

Hypothesis 1. *The PDF f_X of X is monomodal and even (therefore, its only maximum is in zero).*

In most cases Hypothesis 1 is not critical and most of the properties could be easily extended to more general cases. However, Hypothesis 1 simplifies considerably the proofs and it is general enough to be appliable in many real cases (for example, $\mathcal{N}(0, \sigma^2)$ satisfies Hypothesis 1).

Quantization A *uniform quantizer with quantization step* Δ is a function $Q_\Delta : \mathbb{R} \to \mathbb{R}$ that maps every real number $x \in \mathbb{R}$ to the closest multiple of Δ, that is

$$Q_\Delta(x) = \Delta \lfloor x/\Delta \rceil \tag{3}$$

where $\lfloor x \rceil$ denotes the integer nearest to $x \in \mathbb{R}$. We will denote with $\varepsilon_\Delta(x) := x - Q_\Delta(x)$ the *quantization error*. Note that $\varepsilon_\Delta(x) \in I_\Delta$ where I_Δ is defined as

$$I_\Delta := \{u \in \mathbb{R} : -\Delta/2 \le u < \Delta/2\} \tag{4}$$

Sometimes it will be useful to consider the *normalized quantization error*

$$\overline{\varepsilon_\Delta}(x) := \frac{1}{\Delta}\varepsilon_\Delta(x) \in I_1 \tag{5}$$

If X is a random variable, we will denote with X_Δ its quantized version, that is, $X_\Delta := Q_\Delta(X)$

Entropy Let X be a random variable assuming values in a *numerable alphabet* $\mathscr{A}$. A measure of the *randomness* of X, commonly used in cryptography, is its *min entropy* $H_\infty(X)$ defined as [35, 36]

$$H_\infty(X) := \min_{a\in\mathscr{A}}(-\log_2 P[X=a]) = -\log_2 \max_{a\in\mathscr{A}} P[X=a] \tag{6}$$

Min-entropy shares several properties with the Shannon entropy: it attains its maximum value $\log_2|\mathscr{A}|$ if and only if X is uniformly distributed and its minimum value 0 if and only if X is deterministic, if X and Y are independent, then $H_\infty(XY) = H_\infty(X) + H_\infty(Y)$, if g is any function defined on $\mathscr{A}$, then $H_\infty(g(X)) \leq H_\infty(X)$ where equality is guaranteed if g is injective. While Shannon entropy is a measure of the amount of information necessary to describe X, the min entropy is related with the problem of guessing X. For example, it is easy to show that the number of trials necessary to guess X with probability p_{g} cannot be smaller than $p_{\mathrm{g}} 2^{H_\infty(X)}$.

Let X be a continuous r.v. (that is, it assumes values in $\mathbb{R}$) with PDF $f_X : \mathbb{R} \to \mathbb{R}$. We define the *differential min-entropy* of X, $h_\infty(X)$, as

$$h_\infty(X) := -\log_2 \sup_{x\in\mathbb{R}} f_X(x) \tag{7}$$

Some properties of the differential min-entropy will be necessary in the following.

Property 1. *Suppose X is a continuous r.v.*

1. *If $Y = aX + b$, then*
$$h_\infty(Y) = \log_2 a + h_\infty(X) \tag{8}$$
2. *If the support of X is contained in the interval $[a,b]$ (that is, $f_X(x) \neq 0 \Rightarrow a \leq x \leq b$), then*
$$h_\infty(X) \leq \log_2(b-a) \tag{9}$$
with the equality achieved if $X \sim \mathscr{U}([a,b])$.
3. *If f_X satisfies Hypothesis 1, then*
$$H_\infty(X_\Delta) = h_\infty(X) - \log_2\Delta + \alpha(\Delta/\sigma_V) \tag{10}$$
where $\alpha(x) \geq 0$ and $\lim_{x\to 0}\alpha(x) = 0$.

The proof is in Appendix A, Proof A.1.

Remark 2.1

Equation (10) implies that for small Δ (*fine quantization hypothesis*),

$$H_\infty(X_\Delta) \approx h_\infty(X) - \log_2\Delta = h_\infty(\overline{X}) + \log_2\frac{\sigma}{\Delta} \tag{11}$$

where we used $h_\infty(X) = \log_2\sigma + h_\infty(\overline{X})$ (first claim).

3 Random bits from analog sources

3.1 PUCs and APUCs

As said in the introduction, a PUC is a digital circuit whose outcome is very sensitive to component variations happening at construction time. Because of this, it is not possible to predict if a given PUC instance will output '0' or '1'; ideally, approximately half of the PUCs on a chip will produce '0' and the other half will produce '1'.

An APUC, like a PUC, is very sensistive to component variations, but instead of producing a digital output it will produce an *analog* output. Of course, the analog output will need to be *digitalized* in order to have a final digital outcome. For example, in [27] the authors propose an APUC that exploits the dispersion of an *n-well resistor*. By using an operational amplifier the resistance is converted to a tension which is, in turn, converted to digital with an Analog to Digital Converter (ADC).

The advantage of an APUC is that it has the potential of providing more than one bit per cell (up to 6–8 with the APUC of [27], see Section 6). On the other hand, the analog output lacks the intrinsic stability that digital outputs have and this means that we need special care to guarantee that every time an APUC is queried the same digital outcome is produced.

3.2 A generic APUC-based scheme

Fig. 1 shows the reference scheme that we will use in this paper.

- We have an array of *APUC cells*. In an ideal, noiseless case, the ℓ-th cell, when turned on, produces always the same value V_ℓ which is modeled as a random value whose realization is determined at construction time. We will call V_ℓ the *noiseless outcome* of the cell. We will often use the notation V when the cell index can be omitted.

- Noiseless outcome V is modeled as a random variable with PDF f_V, mean m_V and variance σ_V^2. Random variables relative to different cells are iid.

- In practice, because of noise, the value read from a cell will differ from the corresponding noiless outcome V. More precisely, we assume that V gets corrupted by an additive noise $\mathfrak{N}$ that can be used to model, beyond thermal noise, even the effect of other disturbance (e.g., temperature changes [27]). Noise $\mathfrak{N}$ will be assumed to have PDF $f_{\mathfrak{N}}$, zero mean and variance $\sigma_{\mathfrak{N}}^2$.

- The noisy value $\hat{V} = V + \mathfrak{N}$ is converted to digital using an ADC modeled with a quantizer with quantization step Δ. Let $\hat{V}_\Delta = Q_\Delta(\hat{V}) = Q_\Delta(V + \mathfrak{N})$ be the random variable associated with the ADC

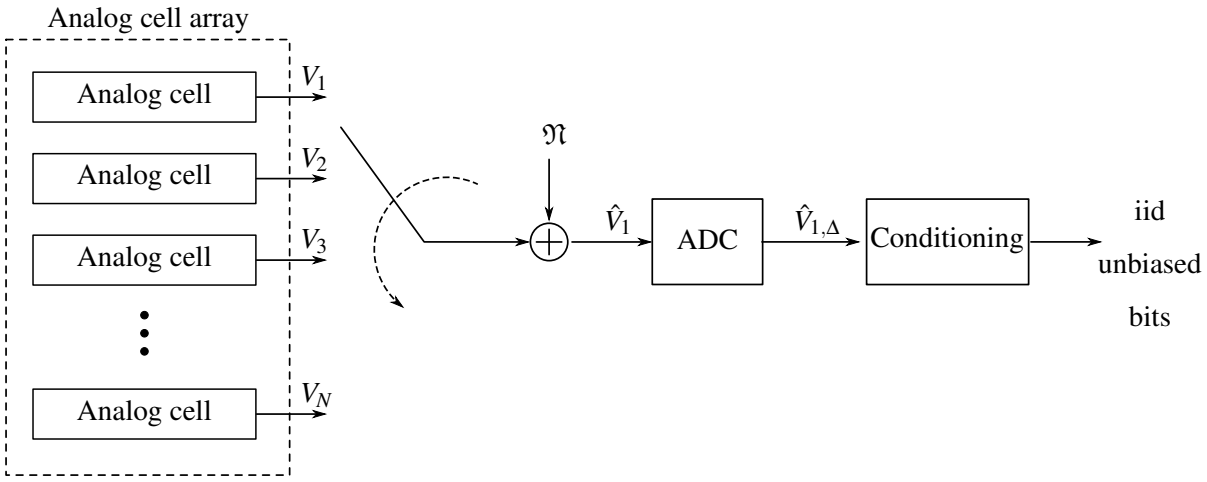

Figure 1: From random analog constants to iid bits.

output.

The quantization of the *noiseless* value, that is, $V_\Delta = Q_\Delta(V)$ will be considered the *reference value* of the cell and we will say that an *error* occurs if $V_\Delta \neq \hat{V}_\Delta$. A cell will be said to be *reliable* if $V_\Delta = \hat{V}_\Delta$ with large enough probability (this idea will be made more precise later).

Note that V_Δ plays only a theoretical role and we do not need to actually know it, although, in practice, it can be needed during the *enrollment* phase, see Section 3.3.

- The quantizer output $\hat{V}_\Delta$ is processed with a *conditioner* which distillates the randomness in $\hat{V}_\Delta$ to produce unbiased iid bits. Since the conditioner is a deterministic function, as long as no error occurs, that is $V_\Delta = \hat{V}_\Delta$ for every cell, the output of the conditioner will be correct.

3.3 The enrollment phase

In many PUF/PUC schemes the first turn-on has a special significance and it is used to do the so-called *enrollment*. Depending on the PUF/PUC scheme, the enrollment phase is used to register the device behavior in a database or creating some *helpers* used in successive turn-on [37, 38]. For example, if the secret generated by a PUC is used as the private key in a DH authentication scheme, during the enrollment phase the public key would be generated and stored together with the device identity.

In the following we will assume that it is possible to get, during enrollment, V_Δ, that is, the quantized version of the noiselss outcome. If this is possible or not, clearly depends on the specific analog cell, but in general one can expect to be able to minimize the noise by averaging many measures (to reduce the impact

of thermal noise) and by a strict control of the environment (e.g., the temperature) to reduce the impact of external variables.

3.4 Conditioning and reliability in an APUC

In order to get a reliable source of random bits from the scheme of Fig. 1 it is necessary to solve two problems

Conditioning That is, exctract from the quantized values $\hat{V}_{\ell,\Delta}$ a sequence of unbiased iid bits.

Reliability That is, guarantee that always the same bit sequence is generated (at least with large enough probability), despite noise $\mathfrak{N}$.

Section 4 describes the proposed conditioning solution, while Section 5 explains how reliability can be achieved. Finally, an example of design for the APUC of [27] is given in Section 6.

4 Conditioning

The conditioning problem is how to convert quantized values $\hat{V}_{\ell,\Delta}$ into a sequence of bits that is, ideally, unbiased and iid or, at least, with some guarantees about its min entropy. An *off–the–shelf* solution is to interpret the sequence of the quantized samples can be seen as simply a sequence of binary words that can be processed by any general conditioning algorithms such as Elias-like schemes [28–34] or simpler solutions such as hash functions [35]. While Elias-like schemes guarantee that the output bit-stream is a perfect Unbiased Bernoulli Process (UBP), they can be more complex to implement if compared with hash-based schemes that, however, do not have the same hard theoretical guarantees, although they can be good enough for most applications. Also note that the original Elias' scheme [28] is inefficient since it produces on average $p_0 - p_0^2$ output bits per input bit, where p_0 is the probability of '0'; even with an UBP (i.e., $p_0 = 1/2$), the efficiency is only 25%.

However, the fact that the conditioner input are quantized analog values enables the use of conditioning schemes especially suited for this type of data such as the Elias-like scheme for Gaussian variables described in [30]. That scheme guarantees a perfect UBP, but it is quite complex to implement. Here we describe a scheme that takes advantage of the analog nature of the data, it is very simple to implement and it has some strong quantitative result about the min-entropy of the final bit sequence.

4.1 Quantizing the quantization error

The proposed approach is quite simple and it exploits the fact that if Δ is small enough (say, less or equal than a threshold Δ_T), the quantization error is approximately uniform. Therefore, one can first quantize V with step Δ_T, take the quantization error $\varepsilon_{\Delta_T}(V) = V - Q_{\Delta_T}(V) \sim \mathscr{U}(I_{\Delta_T})$ and quantize it again with the smaller step $\Delta = 2^{-b}\Delta_T$. Since $\varepsilon_{\Delta_T}(V)$ is (approximately) uniform, the quantized value will be a b-bit binary word with bits (approximately) iid and unbiased. Clearly, instead of doing two successive quantization, it is equivalent (and simpler) to quantize directly V with step Δ and keep only the b least significant bits.

This scheme is very simple, but has as a drawback a loss of efficiency since the result of quantizing with Δ_T (the discarded more significant bits) has some randomness that could be extracted. This will be discussed in Section 4.2.

It is worth to quantify, in terms of min entropy, both the quality of the uniform approximation and the amount of randomness in the most significant bits.

4.1.1 Departure from uniformity of ε_{Δ_T}

Since ε_{Δ_T} has support I_{Δ_T}, according to Property 1, its differential entropy cannot be larger than $\log_2 \Delta_T$, the differential entropy of $\mathscr{U}(I_{\Delta_T})$, a r.v. uniformly distributed over I_{Δ_T}. This suggests to use $\log_2 \Delta_T - h_\infty(\varepsilon_{\Delta_T}) > 0$ as a measure of departure of $f_{\varepsilon_{\Delta_T}}$ from an uniform distribution. An interesting result is that this difference depends on Δ_T and σ_V only via their ratio $\rho = \Delta_T/\sigma_V$, as claimed by the following property whose proof is in Appendix A, Proof A.2.

Property 2. *The differential min entropy of ε_{Δ_T} can be written as*

$$h_\infty(\varepsilon_{\Delta_T}) = \log_2 \Delta_T + h_\infty(\overline{\varepsilon_{\Delta_T}}) \tag{12}$$

where $h_\infty(\overline{\varepsilon_{\Delta_T}})$, the differential entropy of the normalized quantization error, depends on Δ_T and σ_V only via their ratio $\rho = \Delta_T/\sigma_V$, that is,

$$h_\infty(\overline{\varepsilon_{\Delta_T}}) = -\mathfrak{g}_V(\rho) \le 0 \tag{13}$$

for a suitable function $\mathfrak{g}_V$.

We decided to use the minus sign in (13) since in this way $\mathfrak{g}_V(\rho) = \mathfrak{g}_V(\Delta_T/\sigma_V) \ge 0$ can be interpred as the number of bits "lost" because of the departure of ε_{Δ_T} from exact uniformity. Since $\mathfrak{g}_V(\rho)$ depends only on the ratio ρ, we can do plots like the ones in Fig. 2, relative to the Gaussian case and discussed in greater detail in Section 4.3.

An interesting result is that $h_\infty(\varepsilon_{\Delta_T})$ does not depend on the mean, as claimed by the following corollary.

Corollary 1. *Let $U = V + m$, $m \in \mathbb{R}$, and let $\eta_{\Delta_T} = U - Q_{\Delta_T}(U)$ be the corresponding quantization error. For every Δ_T the following holds*

$$h_\infty(\eta_{\Delta_T}) = h_\infty(\varepsilon_{\Delta_T}) \tag{14}$$

The proof is immediate by observing that V and U in Corollary 1 have the same variance.

4.2 Extracting randomness from the most significant part V_{Δ_T}

4.2.1 Entropy of the most significant part V_{Δ_T}

It is useful to know also the min-entropy $H_\infty(V_{\Delta_T})$ of V quantized with step $\Delta_T = 2^b\Delta$ since it represents the number of bits lost when the most significant bits of V_Δ are discarded. Interestingly, even $H_\infty(V_{\Delta_T})$ depends on Δ_T and σ_V only via their ratio. This is a direct consequence of 1 and (10) which imply that

$$H_\infty(V_{\Delta_T}) = h_\infty(\overline{V}) - \log_2(\Delta/\sigma_V) + \alpha(\Delta/\sigma_V) \tag{15}$$

where $\alpha(x) \geq 0$ and $\lim_{x\to 0}\alpha(x) = 0$. Since $H_\infty(V_{\Delta_T})$ depends only on the ratio $\rho = \Delta/\sigma_V$; we can plot graphs like the one in Fig. 2, relative to the Gaussian case and discussed in greater detail in Section 4.3.

4.2.2 Extraction techniques

If the efficiency loss due to discarding the most significant bits is not acceptable (that is, if $H_\infty(V_{\Delta_T})$ is too large), it is possible to extract randomness from the most significant part V_{Δ_T}. In this section we describe and compare two possible approaches, namely: using the sign bit and using a *zero flag*. We V satisfies Hypothesis 1. Note that Hypothesis 1 implies $m_V = \mathbb{E}[V] = 0$; the case $m_V \neq 0$, maybe the most probable case of deviation from Hypothesis 1, is briefly discussed in Section 4.2.3.

Sign bit If the PDF of V is even, the probability of the event $V_{\Delta_T} > 0$ is equal to the probability of the event $V_{\Delta_T} < 0$. This suggests that the sequence of sign bits (1 if V_{Δ_T} is negative) is a UBP. This is true if events $V_{\Delta_T} = 0$ are discarded since otherwise "0" would be more probable than "1." Clearly, the average number of bits per device that can be obtained is (if f_V satisfies Hypothesis 1)

$$P[V_{\Delta_T} \neq 0] = 1 - P[V_{\Delta_T} = 0] = 1 - 2^{-H_\infty(V_{\Delta_T})} = 1 - \rho 2^{-h_\infty(\overline{V})}2^{-\eta(\rho)} \approx 1 - \rho 2^{-h_\infty(\overline{V})} \tag{16}$$

where the approximation holds in the large bitrate limit.

Zero flag Since f_V has a single maximum in zero, it is easy to prove 0 is the most probable value for V_{Δ_T} and $H_\infty(V_{\Delta_T}) = -\log_2 P[V_{\Delta_T} = 0]$. Consider now the *zero flag* r.v. Z equal to 0 if $V_{\Delta_T} = 0$ and equal to 1 otherwise. Clearly, since $P[Z = 0] = P[V_{\Delta_T} = 0]$, if $P[V_{\Delta_T} = 0] \geq 1/2$, then $P[Z = 0] \geq P[Z = 1]$ and

$$H_\infty(Z) = -\log_2 P[V_{\Delta_T} = 0] = H_\infty(V_{\Delta_T}) \tag{17}$$

This shows that if Δ_T is large enough to have $P[V_{\Delta_T} = 0] \geq 1/2$, then the zero flag is optimal since its min-entropy is equal to the min-entropy of V_{Δ_T} and no deterministic processing cannot increase the min-entropy.

4.2.3 The non-zero mean case

The sign method and the zero flag method suppose that f_V is even and it has a single maximum in 0, for example, a zero mean Gaussian. If $m_V \neq 0$, the two methods can be suboptimal. Depending on the context, one can accept the loss of efficiency or estimate the mean m_V and compensate for it; since it is reasonable to assume that in a real case more than one cell will be available, m_V can be estimated by taking the average of the outcomes of the cells. Observe that an error in the estimate of m_V will make the scheme inefficient, but it would not be a critical flaw since it would impact at most one bit of the generated secret.

4.3 The Gaussian case

It is worth specializing the results above to the case $V \sim \mathcal{N}(0, \sigma_V^2)$.

Fig. 2a shows the measure of departure from uniformity of ε_Δ, $\mathfrak{g}_V(\rho) = h_\infty(\varepsilon_\Delta) - \log_2 \Delta$, as function of $\rho = \Delta/\sigma_V$. Notice that for ρ smaller than ≈ 1.7, the quantization error can be considered uniform. This is confirmed by Fig. 2b which shows the PDF of the normalized quantization error $\overline{\varepsilon_\Delta} = \varepsilon_\Delta/\Delta$ for different values of ρ (the normalization allows us to have the horizontal axis in the range $[-0.5, 0.5]$ for every Δ). The inset shows a detail for the case $\rho = 1.35$ that has a special significance in relation to extracting randomness from the most significant part, as it will be clear shortly.

Fig. 2c shows the min-entropy $H_\infty(V_{\Delta_T})$ of the coarse quantization V_{Δ_T} as a function of ρ, compared with the min-entropy achievable with the sign bit or the zero flag approaches. The dashed vertical line corresponds to $\rho = \rho_{\text{opt}} \approx 1.35$. It is clear that in this case the zero flag technique is the optimal one; indeed, for $\rho = \rho_{\text{opt}}$, $P[V_{\Delta_T} = 0] = 0.5$, making the sequence of the zero flags an UBP.

At $\rho = \rho_{\text{opt}}$, the quantization error is well approximated by a uniform random variable. Indeed, according to Fig. 2a, shows that with the difference $h_\infty(\varepsilon_\Delta) - \log_2 \Delta$ at $\rho \approx 1.35$ is approximately 10^{-4}, see also the inset in Fig. 2b.

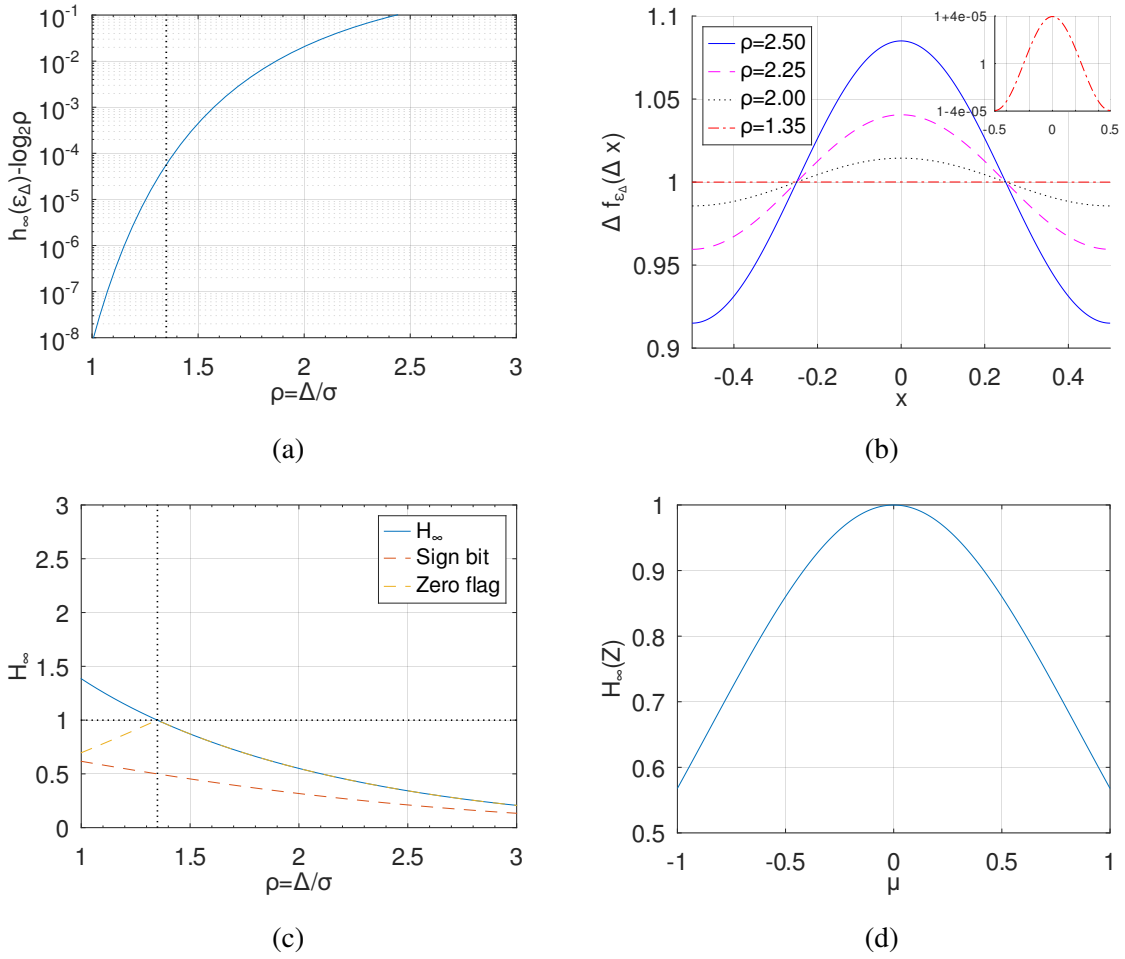

Figure 2: (a) Difference between the differential min entropy of the quantization error of $\mathcal{N}(0,\sigma^2)$ quantized with step Δ and the maximum achievable differential min entropy $-\log_2\Delta$ (b) PDF of the quantization error of normalized quantization error $\overline{\varepsilon_\Delta} = \varepsilon_\Delta/\Delta$ relative to a $\mathcal{N}(0,\sigma^2)$ quantized with step Δ for several values of $\rho = \Delta/\sigma$. The inset shows the case $\rho = 1.35$ with the vertical axis rescaled. (c) Min-entropy of a $\mathcal{N}(0,\sigma^2)$ quantized with step Δ compared with the min-entropy achievable with using the sign bit or the zero flag conditioning. The dotted vertical line corresponds to $\rho = 1.35$ (d) $H_\infty(Z)$ as function of $\mu = m_V/\sigma_V$

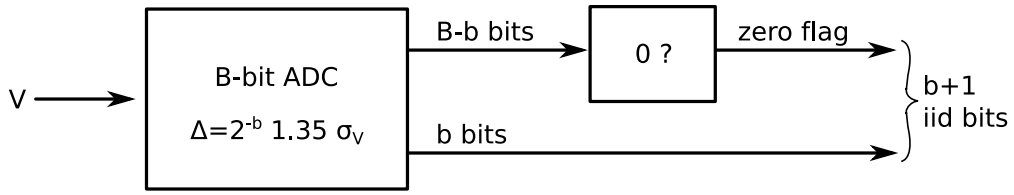

Figure 3: Conditioning Gaussian variables

4.3.1 The non-null mean case

Given its simplicity and efficiency, assume that the zero flag method is used with $\rho = 1.35$. We want to estimate the inefficiency induced in the zero flag method if $m_V \neq 0$. Call $\mu = m_V/\sigma_V$. The probability of $Z = 1$ is

$$P[Z=1] = P[V_\Delta = 0] = \Phi(\rho/2-\mu) - \Phi(-\rho/2-\mu) \tag{18}$$

Define, for convenience of notation,

$$r_\rho(\mu) = \Phi(\rho/2-\mu) - \Phi(-\rho/2-\mu) \tag{19}$$

We will need the following lemma about r_ρ. The proof is in Appendix A, Proof A.3.

Lemma 1. *For every ρ map $\mu \mapsto r_\rho(\mu)$ is even and monotone decreasing for $\mu > 0$.*

Since for $\rho = 1.35$ and $m_V = \mu = 0$, $P[Z=1] = r_{1.35}(0) = 1/2$, it follows that $r_{1.35}(\mu) < 1/2$ when $\mu \neq 0$; therefore, if $\mu \neq 0$, event $Z = 0$ is more probable than $Z = 1$ and this implies

$$H_\infty(Z) = -\log_2 P[Z=0] = -\log_2 r_{1.35}(\mu) \tag{20}$$

that is, $H_\infty(Z)$ for $\rho = 1.35$ depends only on μ. Fig. 2d shows $H_\infty(Z)$ as function of μ.

4.3.2 A recipe for conditioning Gaussian variables

We can summarize the theory so far in the scheme of Fig. 3. Random variable V is quantized with a M-bit ADC and quantization step $\Delta = 2^{-b}\Delta_T = 2^{-b}1.35\,\sigma_V$ for some $b \in \mathbb{N}$. The $M-b$ most significant bits are extracted and used to determine the zero flag that will be equal to 1 if the $M-b$ most significant bits are zero and 0 otherwise. The remaining least significant b bits are taken as they are since they can be considered unbiased and iid with excellent approximation (see Fig. 2). For every analog value $b+1$ bits are produced; the min entropy is between b and $b+1$, depending on $\mu = m_V/\sigma_V$ (the maximum value $b+1$ is achieved for $\mu = 0$, see Fig. 2d).

The value of M can be chosen by fixing a maximum probability of overflow during the conversion, that is, fixing a maximum probability to have

$$|V| > 2^M \Delta = 2^{M-b-1} 1.35 \sigma_V \tag{21}$$

If $M-b=2$, the probability of (21) is $\approx 7 \cdot 10^{-3} = 0.7\%$, while if $M-b=3$, the probability of is $\approx 6.6 \cdot 10^{-8}$. We expect that $M-b=3$ is enough for most application, but maybe even $M-b=2$ can be acceptable in some cases. If the value of $|V|$ is too large, the cell can be disabled, using the mechanism described in Section 5.

5 Reliability

As discussed in Section 4, we can extract as many bit as we desire from V, as long as we choose Δ small enough. Actually, according to (11), Δ gets exponentially small with the number of bits b. See, for example, the Gaussian case in Section 4.3.2.

Beside the difficulties in designing and building an ADC with a large number of bits (schemes like sigma-delta cannot be used in this case), a clear limit to employing a small Δ is the presence of noise $\mathfrak{N}$. For example, if Δ is comparable (or smaller) than $\sigma_{\mathfrak{N}}$, we expect that the least ssignificant bits will change at every measurement and this suggest that the PUC will not be reliable, in the sense that the outcome can change at every turn-on.

One could try to search for the smallest Δ that guarantees a given error probability. Unfortunately this is not possible since, intuitively, values that are close to the border between two quantization intervals are bounded to have an error probability close to $1/2$ (this will be made more precise later). The best we can do is to ask that the probability of having such *irremediably unreliable cells* is small and, as it will be proved later, this can be achieved by choosing Δ large enough.

Here we propose a simple approach similar to the *dark bit* idea [8, 9, 24, 39–44]; that is, at enrollment time we estimate the reliability of every cell and disable the cells that are not reliable enough. This will require to use a surplus of cells in order to compensate for the loss due to the disabled cells. The way the reliability of a cell is checked depends, clearly, on the specific cell architecture.

5.1 Cell reliability definition

We need a precise definition of reliability. Remember that $V_\Delta = Q_\Delta(V)$ is the reference value (the PUC outcome in the noiseless case) and $\hat{V}_\Delta = Q_\Delta(V+\mathfrak{N})$ is the output of the ADC affected by noise. If $\hat{V}_\Delta \neq V_\Delta$ we will say that an *error* occurred. We define the *reliability* of an instance as the complementary of error

probability. More precisely, we define the *reliability function* $\mathscr{R}_\Delta : \mathbb{R} \to [0,1]$ as

$$\mathscr{R}_\Delta(v) := P[Q_\Delta(v+\mathfrak{N}) = Q_\Delta(v)] \tag{22}$$

We will call $\mathscr{R}_\Delta(V)$ the *reliability of the instance* and if $R_{\min}$ is the minimum acceptable reliability, we will say that the instance is *reliable* if $\mathscr{R}_\Delta(V) \geq R_{\min}$.

Remark 5.1

The value of $R_{\min}$ can be expected to be a parameter design or being readly derivable from other parameters. For example, suppose we want to design a PUC with N cells and it is required that the PUC has a "bad start" (that is, it produces the wrong secret) with a probability smaller than $p_{\text{bad on}}$. This would require that reliability of every cell is not smaller than

$$R_{\min} = \sqrt[N]{1 - p_{\text{bad on}}} \tag{23}$$

The following lemma (whose proof is in Appendix A, Proof A.4) simplifies the analysis, showing that $\mathscr{R}_\Delta(v)$ depends only on the quantization error $\varepsilon_\Delta(v)$.

Property 3. *Probability $\mathscr{R}_\Delta(v)$, $v \in \mathbb{R}$ depends only on $\varepsilon_\Delta(v)$, that is, if $v,u \in \mathbb{R}$ are such that $\varepsilon_\Delta(v) = \varepsilon_\Delta(u)$, then $\mathscr{R}_\Delta(v) = \mathscr{R}_\Delta(u)$. Moreover, if $\mathfrak{N}$ has zero mean and variance $\sigma^2_{\mathfrak{N}}$, then for every $|v| \leq \Delta/2$*

$$\mathscr{R}_\Delta(v) = F_{\mathfrak{N}}\left(\frac{\Delta}{2} - v\right) - F_{\mathfrak{N}}\left(-\frac{\Delta}{2} - v\right) \tag{24}$$

Remark 5.2

It is interesting to consider the normalized version

$$\overline{\mathscr{R}}_\kappa(x) := \mathscr{R}_\Delta(\Delta x) = F_{\overline{\mathfrak{N}}}\left(\kappa\left(\frac{1}{2} - x\right)\right) - F_{\overline{\mathfrak{N}}}\left(\kappa\left(-\frac{1}{2} - x\right)\right) \qquad |x| < 1/2 \tag{25}$$

where $F_{\overline{\mathfrak{N}}}$ is the CDF of the normalized noise $\overline{\mathfrak{N}} = \mathfrak{N}/\sigma_{\mathfrak{N}}$. Note that $\overline{\mathscr{R}}_\kappa(x)$ depends only on the ratio $\kappa = \Delta/\sigma_{\mathfrak{N}}$. This allows plots like the one in Fig. 4 that shows $1 - \overline{\mathscr{R}}_\kappa(x)$ in the Gaussian case for different values of κ.

In Fig. 4 it is clear that $\overline{\mathscr{R}}_\kappa(x)$ is maximum when $x = 0$, it increases with κ and it is always near $1/2$ (actually, smaller than $1/2$) on the extreme values $x = \pm 1/2$, confirming the intuition that the values near the border between quantization steps are bound to be unreliable. The following property, shows that this is a general consequence of the symmetry of the PDF of the noise, therefore it holds in many cases of interest.

Property 4. *If the density of $\overline{\mathfrak{N}}$ satisfies Hypothesis 1, then*

1. *$\overline{\mathscr{R}}_\kappa$ is even,*
2. *For every κ, $\overline{\mathscr{R}}_\kappa(1/2) \leq 1/2$*
3. *For every κ, map $x \to \overline{\mathscr{R}}_\kappa(x)$ is monotone decreasing for $x > 0$*

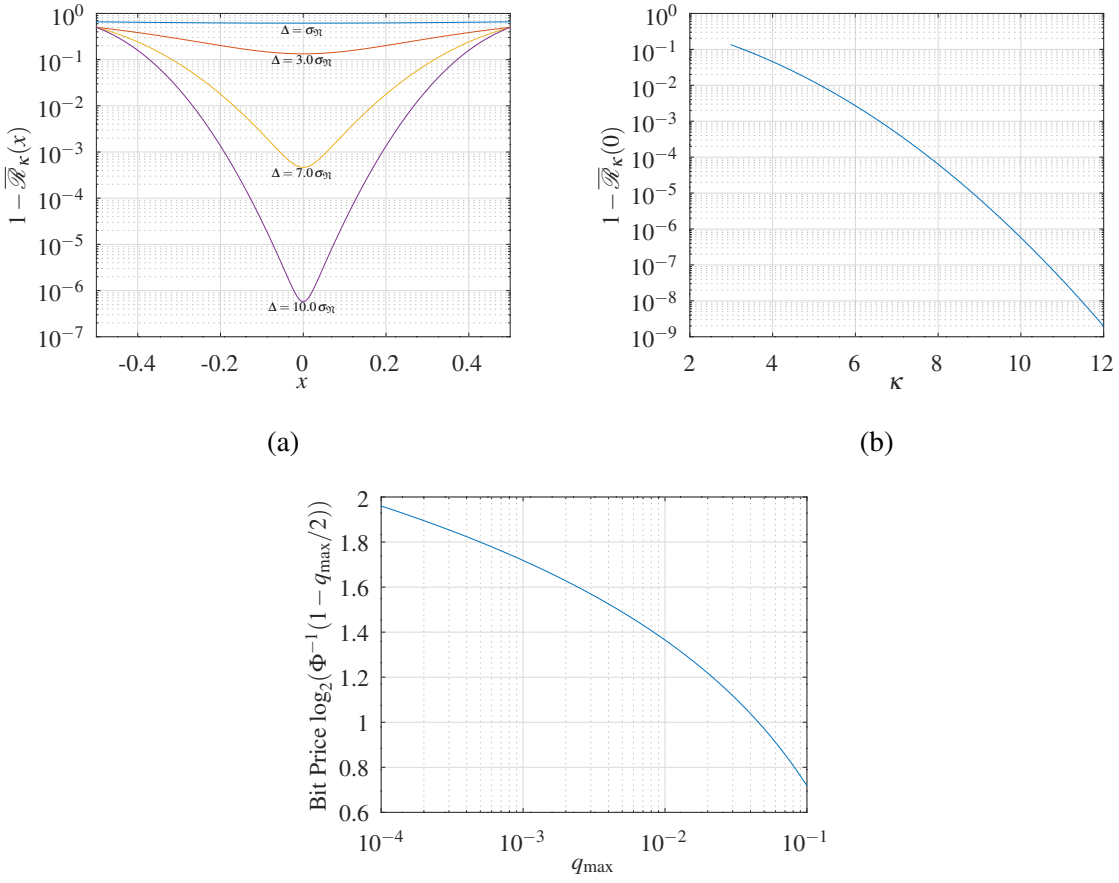

Figure 4: (a) Error probability function for the Gaussian case and few values of $\kappa = \Delta/\sigma_{\mathfrak{N}}$ (b) Minimum error probability achievable as a function of κ

4. *For every $|x| \le 1/2$ map $\kappa \to \overline{\mathscr{R}}_\kappa(x)$ is monotone increasing.*

The proof is in Appendix A, Proof A.5. Property 4 has two consequences that will prove important.

Corollary 2. *If the hypothesis of Property 4 are satisfied, then for all $v \in [-1/2, 1/2]$*

$$\overline{\mathscr{R}}_\kappa(0) = 2F_{\overline{\mathfrak{N}}}(\kappa/2) - 1 \ge \overline{\mathscr{R}}_\kappa(v) \ge \overline{\mathscr{R}}_\kappa(1/2) = F_{\overline{\mathfrak{N}}}(0) - F_{\overline{\mathfrak{N}}}(-\kappa) < 1/2 \tag{26}$$

Definition 1. *If $q \in [\overline{\mathscr{R}}_\kappa(1/2), \overline{\mathscr{R}}_\kappa(0)] = [\mathscr{R}_\Delta(\Delta/2), \mathscr{R}_\Delta(0)]$, we will denote with $\mathscr{R}_\Delta^{-1}(q) \in [0, \Delta/2]$ the positive solution of the equation $\mathscr{R}_\Delta(x) = q$. The definition is well posed since $\mathscr{R}_\Delta(x) = q$ has one and only one positive solution, as a consequence of Property 4 and Corollary 2.*

Remark 5.3 (Bound on κ)

Since $\overline{\mathscr{R}}_\kappa(0)$ is a monotone decreasing function of κ, it follows that if a minimun reliability $R_{\min}$ is desired, κ must be not smaller than

$$\kappa_{\min} = 2F_{\overline{\mathfrak{N}}}^{-1}\left(\frac{1+R_{\min}}{2}\right) \tag{27}$$

The lower bound on $\kappa = \Delta/\sigma_{\mathfrak{N}}$ implies a lower bound on $\Delta = 2^{-b}\Delta_T$ which in turn implies an upper bound of b. See also Fig. 4b.

5.2 Probability of a reliable cell

Because of Property 4 and Corollary 2 above, we can map the reliability requirement into a constraint about $\varepsilon_\Delta(V)$. More precisely, if $R_{\min} > \mathscr{R}_\Delta(0)$, the constraint is never satisfied since the instance reliability will always be smaller then $R_{\min}$. If $R_{\min} < \mathscr{R}_\Delta(\Delta/2)$, the reliability required is so small that every instance satisfies constraint (23). Finally, in the intermediate case, constraint (23) is equivalent to

$$|\varepsilon_\Delta(V)| \le \mathscr{R}_\Delta^{-1}(R_{\min}) \tag{28}$$

From this result we can derive the probability of having a reliable cell, p_{rel}, as

$$p_{\text{rel}} = \begin{cases} 1 & R_{\min} < \mathscr{R}_\Delta(\Delta/2) \\ \approx 2\overline{\mathscr{R}}_\kappa^{-1}(R_{\min}) & R_{\min} \in [\mathscr{R}_\Delta(0), \mathscr{R}_\Delta(\Delta/2)] \\ 0 & R_{\min} > \mathscr{R}_\Delta(0) \end{cases} \tag{29}$$

where the second case follows from

$$P[|\varepsilon_\Delta(V)| \le \mathscr{R}_\Delta^{-1}(R_{\min})] \approx \frac{2\mathscr{R}_\Delta^{-1}(R_{\min})}{\Delta} = 2\overline{\mathscr{R}}_\kappa^{-1}(R_{\min}) \tag{30}$$

where we used the fine quantization hypothesis, that is, that $\varepsilon_\Delta(V)$ is approximately uniform on $(-\Delta/2, \Delta/2)$.

Fig. 5 shows the probability of getting an *unreliable* cell (that is, the complementary of (29)) as function of the reliability for different values of κ and ξ. We used values of κ and ξ comparable with those found in [27]. Fig. 5 shows the complementary of (29) in order to have a more detailed view of the region where (29) is almost one.

5.3 Dark bit approach

To guarantee the reliability of every cell is not smaller than $R_{\min}$, we propose a *dark bit* approach [8, 9, 24, 39–44], that is, at enrollment time the reliability of every cell is estimated and the cells that are not reliable enough are disabled. How the reliability is estimated and how the cells are disabled depend, clearly, on the specific application and the specific APUC scheme employed and it will not be discussed further here.

If N working cells are required, the chip must include a surplus, say $N^+ > N$ cells, in order to compensate for the disabled cells. We assume that if more than $N^+ - N$ cells are disabled, the chip is discarded. If p_{rel} is the probability of a reliable cell, the probability of discarding a chip is

$$p_{\text{bad}} = P[\mathscr{B}(N^+, p_{\text{rel}}) < N] = \mathfrak{I}_{1-p_{\text{rel}}}(N^+ - N, 1+N) \tag{31}$$

where $\mathscr{B}(N^+, p_{\text{rel}})$ is a *binomial random variable* with N^+ trials and success probability p_{rel} and $\mathfrak{I}$ is the *regurlarized incomplete beta function* [45].

6 Design example

In this section we show how to apply the theory above to a PUC design. We will assume that we are given

1. The number of bits B produced by the PUC
2. The probability of a bad turn-on $p_{\text{bad on}}$.
3. The statistical description of V. For the sake of this example we will assume $V \sim \mathscr{N}(0, \sigma_V^2)$
4. The statistical description of noise $\mathfrak{N}$. For the sake of this example we will assume $\mathfrak{N} \sim \mathscr{N}(0, \sigma_{\mathfrak{N}}^2)$

We employ a *dark-bit* approach: at enrollment time we estimate the reliability of the cell and if it is not reliable enough, we disable it. If the remaining cells can produce at least B, we keep the chip, otherwise we discard it.

Designing the PUC means to find

1. The quantization step $\Delta = 2^{-b}\Delta_T$ (or, equivalently, the number of bits per cell b), where $\Delta_T = \rho_{\text{opt}}\sigma_V = 1.35\sigma_V$ (see Section 4.3).

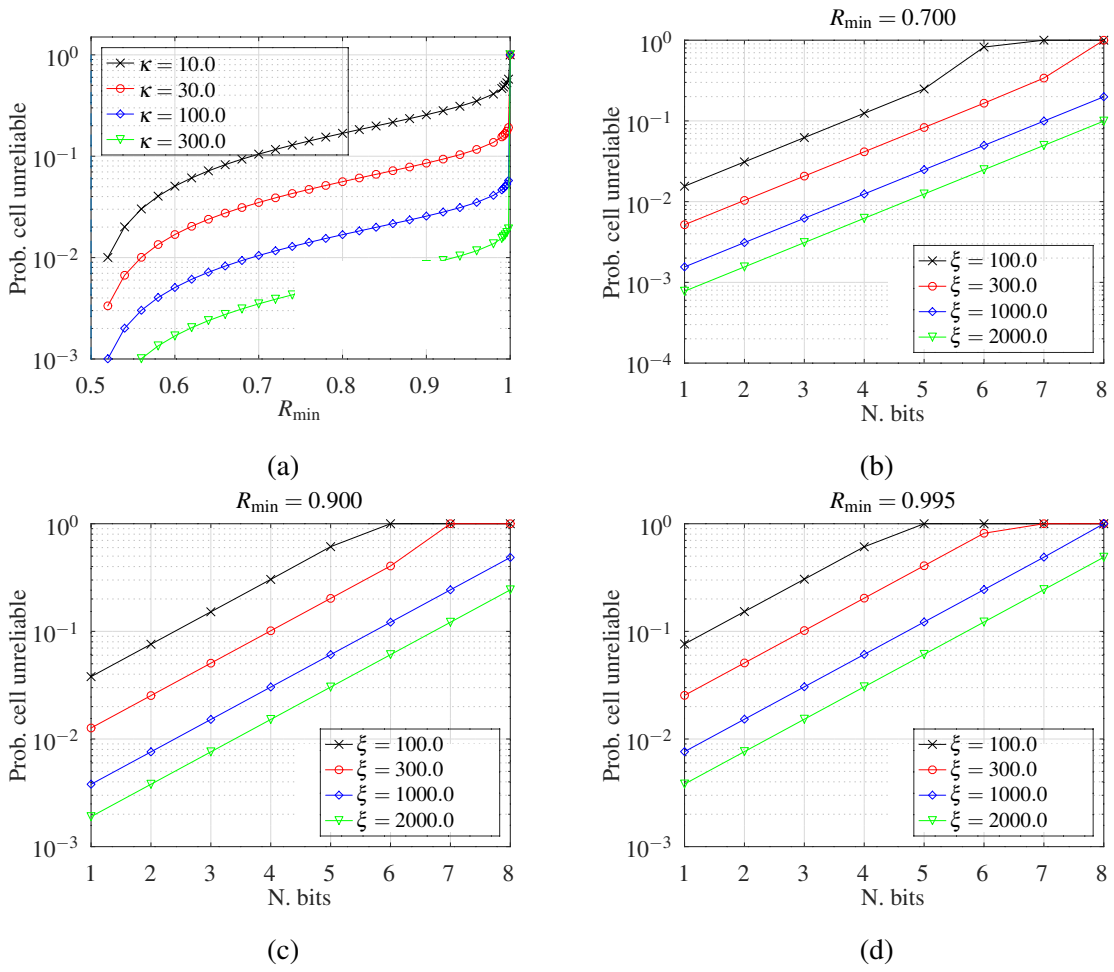

Figure 5: (a) Probability of getting an *unreliable* cell as function of the minimum reliability $R_{\min}$ and κ. The dashed vertical line marks $R_{\min} = 0.5$ (b)-(d) Probability of getting an *unreliable* cell as function of the number of bits per cell b for different values of SNR ξ and minimum reliability $R_{\min}$.

2. The number of cells to be implemented in the chip N^+, remembering that the minimum number of working cells is $N = \lceil B/b \rceil$.

For the sake of this example, we choose $N^+/(1-p_{\text{bad}})$ as the objective function to be minimized. The rationale for this choice is that ratio $N^+/(1-p_{\text{bad}})$ can be interpreted as the average number of cells required to generate B bits.

The outline of a possible design procedure is as follow

1. Choose a number of bits per cell b
2. Compute the required number of working cells $N = \lceil B/b \rceil$ and the minimum reliability $R_{\min}$ required using (23)
3. Compute $\kappa = \Delta/\sigma_{\mathfrak{N}} = 2^{-b}1.35\xi$ and check that $\overline{\mathscr{R}}_\kappa(0) \geq R_{\min}$. If the check fails, b is not a feasible number of bits per cell and the design procedure cannot continue.
4. Compute the reliable cell probability p_{rel} usng (29)
5. Find $N^+ \geq N$ that minimizes

$$\frac{N^+}{1-p_{\text{bad}}} = \frac{N^+}{P[\mathscr{B}(N^+, p_{\text{rel}}) \geq N]} \tag{32}$$

The procedure above is iterated for several choices of b to find the best one, that is, the value of b that gives rise to the smallest $N^+/(1-p_{\text{bad}})$.

Remark 6.1 (Optimal N^+ in (32))

At the best of our knowledge, there is not a simple closed form expression for the optimal N^+ in (32); this is not a critical issue since (32) can be easily optimized by brute force, as soon as p_{rel} and N are known.

Fig. 6 shows the result of said optimization for few values of N and p_{rel}. The vertical axis shows $N^+/(N/p_{\text{rel}}) = p_{\text{rel}}N^+/N$, that is the ratio between the average number of working cells and the required number of working cells.

Fig. 7 shows the total number of bit per cell B/N^+ as function of SNR ξ and for different values of the PUC size B and $p_{\text{bad on}}$. The range of SNRs is comparable with the SNR found in [27].

7 Conclusions

A simple digitizer based on the properties of quantization noise was proposed. The digitizer first processes the analog input value with an ADC and then it discards the most significant bits. Theoretical analysis show that the optimal quantizaztion step Δ is approximately 1.35 times the standard deviation of the input.

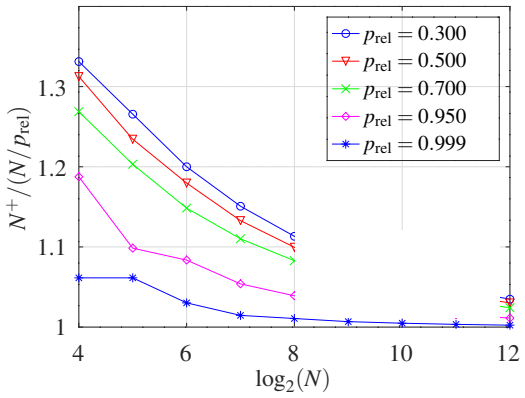

Figure 6: Excess redundancy $N^+/(N/p_{\mathrm{rel}})$ as function of N for different valus of p_{rel}

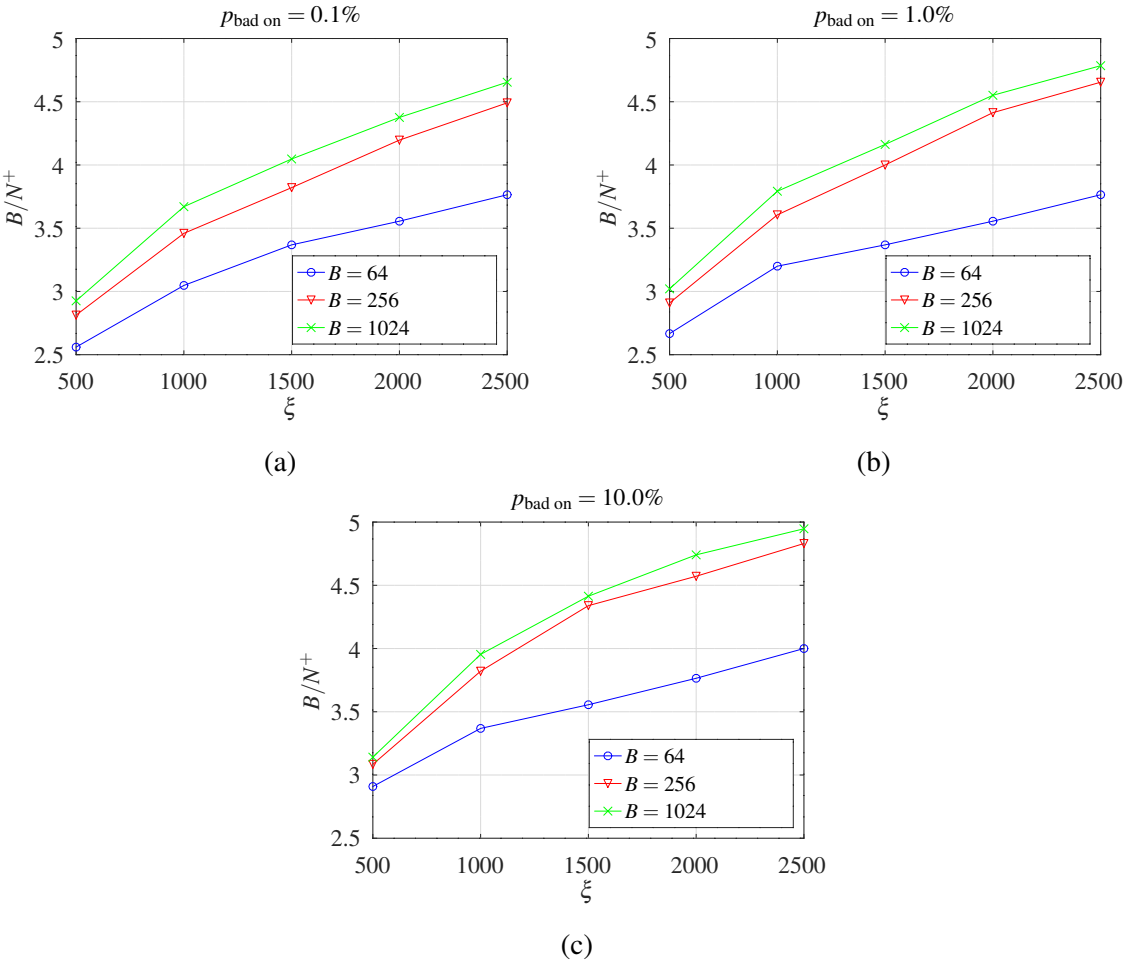

Figure 7: Examples of APUC design

If required, the most significant bits can be processed to distill further random bits. We analyzed two methods to process the most significant bits and it was found out that the most efficient (and easiest) method is the *zero flag method*: if the most significant bits are all zeros, the output is 0, otherwise it is 1. This method is optimal in some cases.

Finally, it was shown that the maximum number of bit per cell that can be extracted is function of the required error probability p_{bad} and the ratio between the standar deviations of the input and of the noise.

References

[1] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in *Proceedings of the 9th ACM conference on Computer and communications security*, CCS '02, (New York, NY, USA), pp. 148–160, ACM, 2002.

[2] A. Herkle, J. Becker, and M. Ortmanns, "An arbiter PUF employing eye-opening oscillation for improved noise suppression," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, May 2018.

[3] D. Lim, J. Lee, B. Gassend, G. Suh, M. van Dijk, and S. Devadas, "Extracting secret keys from integrated circuits," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 13, no. 10, pp. 1200–1205, 2005.

[4] D. Lim, "Extracting secret keys from integrated circuits," Master's thesis, MIT, May 2004.

[5] G. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Design Automation Conference, 2007. DAC '07. 44th ACM/IEEE*, pp. 9–14, 2007.

[6] D. E. Holcomb, W. P. Burleson, and K. Fu, "Initial SRAM state as a fingerprint and source of true random numbers for rfid tags," in *In Proceedings of the Conference on RFID Security*, 2007.

[7] R. Maes, P. Tuyls, and I. Verbauwhede, "A soft decision helper data algorithm for SRAM PUFs," in *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*, pp. 2101–2105, 2009.

[8] S. Satpathy, S. Mathew, V. Suresh, and R. Krishnamurthy, "Ultra-low energy security circuits for IoT applications," in *2016 IEEE 34th International Conference on Computer Design (ICCD)*, pp. 682–685, Oct 2016.

[9] S. Satpathy, S. K. Mathew, V. Suresh, M. A. Anders, H. Kaul, A. Agarwal, S. K. Hsu, G. Chen, R. K. Krishnamurthy, and V. K. De, "A 4-fJ/b Delay-Hardened Physically Unclonable Function Circuit With

Selective Bit Destabilization in 14-nm Trigate CMOS," *IEEE Journal of Solid-State Circuits*, vol. 52, pp. 940–949, April 2017.

[10] A. Alvarez and M. Alioto, *Security Down to the Hardware Level*, pp. 247–270. Cham: Springer International Publishing, 2017.

[11] S. Taneja, A. Alvarez, G. Sadagopan, and M. Alioto, "A fully-synthesizable c-element based puf featuring temperature variation compensation with native 2.8% ber, 1.02fj/b at 0.8–1.0v in 40nm," in *2017 IEEE Asian Solid-State Circuits Conference (A-SSCC)*, pp. 301–304, Nov 2017.

[12] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in *Proceedings of the 1st ACM Conference on Computer and Communications Security*, CCS '93, (New York, NY, USA), pp. 62–73, ACM, 1993.

[13] O. Goldreich, S. Goldwasser, and S. Micali, "How to construct random functions," *J. ACM*, vol. 33, pp. 792–807, Aug. 1986.

[14] K. Matsunaga, S. Oshima, T. Minotani, T. Kondo, and H. Morimura, "Automatic identification number generation circuit using NMOS pair current mismatch," *Japanese Journal of Applied Physics*, vol. 54, no. 4S, p. 04DE12, 2015.

[15] K. Lofstrom, W. Daasch, and D. Taylor, "IC identification circuit using device mismatch," in *Solid-State Circuits Conference, 2000. Digest of Technical Papers. ISSCC. 2000 IEEE International*, pp. 372–373, Feb 2000.

[16] D. E. Holcomb, W. P. Burleson, and K. Fu, "Power-up SRAM state as an identifying fingerprint and source of true random numbers," *IEEE Transactions on Computers*, 2009.

[17] Y. Su, J. Holleman, and B. P. Otis, "A 1.6pj/bit 96% stable chip-id generating circuit using process variations," in *2007 IEEE International Solid-State Circuits Conference, ISSCC 2007, Digest of Technical Papers, San Francisco, CA, USA, February 11-15, 2007*, pp. 406–611, 2007.

[18] E. S. Kumar, J. Guajardo, R. Maesyz, G. jan Schrijen, and P. Tuyls, "Extended abstract: The butterfly PUF protecting IP on every FPGA," in *in Hardware-Oriented Security and Trust. HOST, 2008. IEEE International Workshop on*, pp. 67–70, 2008.

[19] Y. Su, J. Holleman, and B. Otis, "A digital 1.6 pJ/bit chip identification circuit using process variations," *Solid-State Circuits, IEEE Journal of*, vol. 43, pp. 69–77, Jan 2008.

[20] X. Xu, A. Rahmati, D. Holcomb, K. Fu, and W. Burleson, "Reliable physical unclonable functions using data retention voltage of SRAM cells," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2015. doi: 10.1109/TCAD.2015.2418288.

[21] J. Lee, D. Lim, B. Gassend, G. Suh, M. van Dijk, and S. Devadas, "A technique to build a secret key in integrated circuits for identification and authentication applications," in *VLSI Circuits, 2004. Digest of Technical Papers. 2004 Symposium on*, pp. 176–179, June 2004.

[22] P. Prabhu, A. Akel, L. Grupp, W.-K. Yu, G. Suh, E. Kan, and S. Swanson, "Extracting device fingerprints from flash memory by exploiting physical variations," in *Trust and Trustworthy Computing* (J. McCune, B. Balacheff, A. Perrig, A.-R. Sadeghi, A. Sasse, and Y. Beres, eds.), vol. 6740 of *Lecture Notes in Computer Science*, pp. 188–201, Springer Berlin Heidelberg, 2011.

[23] F. Tehranipoor, N. Karimian, W. Yan, and J. A. Chandy, "DRAM-based intrinsic physically unclonable functions for system-level security and authentication," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. PP, no. 99, pp. 1–13, 2016.

[24] R. Bernardini and R. Rinaldo, "Analytic and simulation results about a compact, reliable, and unbiased 1-bit physically unclonable constant," *IEEE Transactions on Information Forensics and Security*, vol. 11, pp. 2804–2817, Dec. 2016.

[25] C. Bai, X. Zou, and K. Dai, "A highly stable R-Diode-based physical unclonable function," in *ICINS 2014 - 2014 International Conference on Information and Network Security*, pp. 22–27, Nov 2014.

[26] R. Bernardini and R. Rinaldo, "A very stable diode-based physically unclonable constant," *Integr. VLSI J.*, vol. 59, pp. 179–189, Sept. 2017.

[27] R. Bernardini, "Analytic and Simulation Results of a Gaussian Physically Unclonable Constant Based on Resistance Dispersion." ArXiv, Mar. 2025.

[28] P. Elias, "The efficient construction of an unbiased random sequence," *Ann. Math. Statist.*, vol. 43, pp. 865–870, 1972.

[29] J. von Neumann, "Various techniques used in connection with random digits," *Appl. Math. Ser., Notes by G. E. Forstyle, Nat. Bur. Stand.*, vol. 12, pp. 36–38, 1951.

[30] R. Bernardini and R. Rinaldo, "Generalized elias schemes for efficient harvesting of truly random bits," *International Journal of Information Security*, vol. 17, pp. 67–81, Feb 2018.

[31] W. Hoeffding and G. Simon, "Unbiased coin tossing with a biased coin," *Ann. Math. Statist.*, vol. 41, pp. 341–352, 1970.

[32] Q. Stout and B. Warren, "Unbiased coin tossing with a biased coin," *Ann. Probab.*, vol. 12, pp. 212–222, 1984.

[33] Y. Peres, "Unbiased coin tossing with a biased coin," *Ann. Statist.*, vol. 20, pp. 590–597, 1992.

[34] H. Zhou and J. Bruck, "Efficient generation of random bits from finite state markov chains," *Information Theory, IEEE Transactions on*, vol. 58, no. 4, pp. 2490–2506, 2012.

[35] E. Barker and J. Kelsey, "Recommendation for the entropy sources used for random bit generation," tech. rep., NIST — Computer Security Division, aug 2012. NIST Special Publication 800-90B.

[36] R. Konig, R. Renner, and C. Schaffner, "The operational meaning of min- and max-entropy," *IEEE Transactions on Information Theory*, vol. 55, no. 9, pp. 4337–4347, 2009.

[37] S. Joshi, S. P. Mohanty, and E. Kougianos, "Everything you wanted to know about pufs," *IEEE Potentials*, vol. 36, no. 6, pp. 38–46, 2017.

[38] C. Herder, M.-D. Yu, F. Koushanfar, and S. Devadas, "Physical unclonable functions and applications: A tutorial," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1126–1141, 2014.

[39] R. Bernardini and R. Rinaldo, "Theoretical limits of helper-less stabilizers for physically unclonable constants," *Emerging Topics in Computing, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2014. doi : 10.1109/TETC.2014.2386137.

[40] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, *FPGA Intrinsic PUFs and Their Use for IP Protection*, pp. 63–80. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.

[41] S. K. Mathew, S. K. Satpathy, M. A. Anders, H. Kaul, S. K. Hsu, A. Agarwal, G. K. Chen, R. J. Parker, R. K. Krishnamurthy, and V. De, "16.2 A 0.19pJ/b PVT-variation-tolerant hybrid physically unclonable function circuit for 100% stable secure key generation in 22nm CMOS," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 278–279, Feb 2014.

[42] S. Satpathy, S. Mathew, J. Li, P. Koeberl, M. Anders, H. Kaul, G. K. Chen, A. Agarwal, S. Hsu, and R. Krishnamurthy, "13fj/bit probing-resilient 250k PUF array with soft darkbit masking for 1.94% bit-error in 22nm tri-gate CMOS," in *ESSCIRC 2014 - 40th European Solid State Circuits Conference, Venice Lido, Italy, September 22-26, 2014*, pp. 239–242, 2014.

[43] S. Mathew, S. Satpathy, V. Suresh, and R. K. Krishnamurthy, "Energy efficient and ultra low voltage security circuits for nanoscale CMOS technologies," in *2017 IEEE Custom Integrated Circuits Conference (CICC)*, pp. 1–4, April 2017.

[44] R. Bernardini and R. Rinaldo, "Analysis of some simple stabilizers for physically obfuscated keys," *International Journal of Information Security*, Oct. 2019.

[45] F. Olver, N. I. of Standards, and T. (U.S.), *NIST Handbook of Mathematical Functions Hardback and CD-ROM*. Cambridge University Press, 2010.

A Proofs

Proof A.1. *Proof of Property 1* The first claim follows at once from $f_Y(x) = (1/a) f_X((x-b)/a)$. To prove the second claim, call $M = \max_{x\in[a,b]} f_X(x)$ the maximum of f_X and observe that

$$1 = \int_{\mathbb{R}} f_X(x)dx = \int_a^b f_X(x)dx \leq M(b-a) \tag{33}$$

which shows that $M \geq 1/(b-a)$. The second claim follows at once.

Finally, to prove the third claim observe that

$$P[X_\Delta = n\Delta] = \int_{I_\Delta + n\Delta} f_X(x)dx \leq \int_{I_\Delta + n\Delta} \left[\max_{u\in\mathbb{R}} f_X(u)\right] dx = \Delta \max_{u\in\mathbb{R}} f_X(u) \tag{34}$$

Moreover, from the hypothesis on f_X it follows that for every $n > 0$

$$\begin{aligned}
P[X_\Delta = n\Delta] &= \int_{n\Delta-\Delta/2}^{n\Delta+\Delta/2} f_X(x)dx \\
&= \int_{n\Delta-\Delta/2}^{n\Delta} f_X(x)dx + \int_{n\Delta}^{n\Delta+\Delta/2} f_X(x)dx \\
&= \int_{n\Delta-\Delta/2}^{n\Delta} f_X(x)dx + \int_{n\Delta-\Delta/2}^{n\Delta} f_X(x+\Delta/2)dx \\
&\leq 2\int_{n\Delta-\Delta/2}^{n\Delta} f_X(x)dx \qquad\qquad f_X(x+\Delta/2) \leq f_X(x) \text{ if } x > 0 \\
&= 2\int_0^{\Delta/2} f_X(x+n\Delta-\Delta/2)dx \\
&\leq 2\int_0^{\Delta/2} f_X(x)dx \\
&= P[X_\Delta = 0]
\end{aligned} \tag{35}$$

and similarly for $n < 0$. It follows that the minimum entropy of X_Δ is $H_\infty(X_\Delta) = -\log_2 P[X_\Delta = 0]$. Observe that $P[X_\Delta = 0]$ can be bounded as follows

$$\Delta f_X(\Delta/2) \leq P[X_\Delta = 0] \leq \Delta f_X(0) \tag{36}$$

which allows us to bound $H_\infty(X_\Delta)$ as

$$-\log_2\Delta-\log_2 f_X(\Delta/2)\geq H_\infty(X_\Delta)\geq-\log_2\Delta-\log_2 f_X(0)=-\log_2\Delta+h_\infty(X) \tag{37}$$

It follows that $H_\infty(X_\Delta)-(h_\infty(X)-\log_2\Delta)$ goes to zero as $\Delta\to 0$. □

Proof A.2. *Proof of Property 2* We have three claims to prove

Claim 1: Observe that

$$f_{\varepsilon_\Delta}(x)=\sum_{n\in\mathbb{Z}}\tau^{n\Delta}f_V(x-n\Delta)=\sum_{n\in\mathbb{Z}}[\tau^{n\Delta}f_V](x)\qquad x\in[-\Delta/2,\Delta/2] \tag{38}$$

Remember that $\overline{\varepsilon_\Delta}=\varepsilon_\Delta/\Delta$, therefore

$$\begin{aligned}f_{\overline{\varepsilon_\Delta}}=S_\Delta f_{\varepsilon_\Delta}&=S_\Delta\sum_{n\in\mathbb{Z}}\tau^{n\Delta}f_V\\&=\sum_{n\in\mathbb{Z}}\tau^n S_\Delta f_V\\&=\sum_{n\in\mathbb{Z}}\tau^n S_\Delta S_{1/\sigma_V}f_{\overline{V}}\\&=\sum_{n\in\mathbb{Z}}\tau^n S_{\Delta/\sigma_V}f_{\overline{V}}\end{aligned} \tag{39}$$

Observe that $f_{\overline{\varepsilon_\Delta}}$ depends only on Δ/σ_V. Call

$$\mathfrak{g}_V(\Delta/\sigma_V):=h_\infty(\overline{\varepsilon_\Delta}) \tag{40}$$

where we emphasized the fact that $h_\infty(\overline{\varepsilon_\Delta})$ depends only on Δ/σ_V. Observe that since the support of $f_{\overline{\varepsilon_\Delta}}=I_\Delta 1$, it must be $\max_{x\in\mathbb{R}}f_{\overline{\varepsilon_\Delta}}\geq 1$, therefore, $\mathfrak{g}_V(\Delta/\sigma_V)=-\log_2\max_{x\in\mathbb{R}}f_{\overline{\varepsilon_\Delta}}\leq 0$.

Claim 2: Observe that since $\Delta\overline{\varepsilon_\Delta}=\varepsilon_\Delta$, then

$$h_\infty(\varepsilon_\Delta)=\log_2\Delta+h_\infty(\overline{\varepsilon_\Delta})=\log_2\Delta+\mathfrak{g}_V(\Delta/\sigma_V) \tag{41}$$

Claim 3: Observe that $f_U=\tau^{-m}f_V$; it follows that $f_{\eta_{\Delta_T}}=\tau^{-m \bmod \Delta}f_{\varepsilon_\Delta}$; therefore the min-entropy is the same. □

Proof A.3. *Proof of Lemma 1* The fact that r_ρ is even follows from

$$\begin{aligned}r_\rho(\mu)&=P[\mathscr{N}\in(-\rho/2-\mu,\rho/2-\mu)]\\&=P[-\mathscr{N}\in(-\rho/2-\mu,\rho/2-\mu)]\\&=P[\mathscr{N}\in(-\rho/2+\mu,\rho/2+\mu,)]=r_\rho(-\mu)\end{aligned} \tag{42}$$

In order to prove the monotonicity for $\mu > 0$, compute the derivative

$$\begin{aligned} r'_\rho(\mu) &= -\phi(\rho/2-\mu)+\phi(-\rho/2-\mu) \\ &= -\phi(\rho/2-\mu)+\phi(\rho/2+\mu) \end{aligned} \tag{43}$$

where ϕ is the PDF of $\mathcal{N}$ and we used $\phi(x) = \phi(-x)$. If $\mu > 0$, then $\rho/2+\mu > \rho/2-\mu$ and from (43) it follows $r'_\rho(\mu) < 0$. □

Proof A.4. *Proof of Property 3* In order to prove the first claim ($\mathscr{R}_\Delta(v)$ depends only on $\varepsilon_\Delta(v)$, write v as $v = n\Delta + \varepsilon_\Delta(v)$ where $n = Q_\Delta(v) \in \mathbb{Z}$. It follows

$$\begin{aligned} \mathscr{R}_\Delta(v) &= P[Q_\Delta(v+\mathfrak{N}) = n\Delta] \\ &= P[Q_\Delta(n\Delta+\varepsilon_\Delta(v)+\mathfrak{N}) = n\Delta] \\ &= P[n\Delta+\varepsilon_\Delta(v)+\mathfrak{N} \in n\Delta + I_\Delta] \\ &= P[\mathfrak{N} \in I_\Delta - \varepsilon_\Delta(v)] \end{aligned} \tag{44}$$

The last term of (44) depends only on $\varepsilon_\Delta(v)$, proving the first claim. Now rewrite the last term of (44) as follows

$$\begin{aligned} \mathscr{R}_\Delta(\varepsilon_\Delta) = P[\mathfrak{N} \in I_\Delta - \varepsilon_\Delta] &= P[-\Delta/2-\varepsilon_\Delta \le \mathfrak{N} < \Delta/2-\varepsilon_\Delta] \\ &= F_\mathfrak{N}\left(\frac{\Delta}{2}-v\right) - F_\mathfrak{N}\left(-\frac{\Delta}{2}-v\right) \end{aligned} \tag{45}$$

□

Proof A.5. *Proof of Property 4 Claim 1:* Since $f_{\overline{\mathfrak{N}}}$ is even, then the CDF enjoys the symmetry $F_{\overline{\mathfrak{N}}}(x) = F_{\overline{\mathfrak{N}}}(x) = 1 - F_{\overline{\mathfrak{N}}}(-x)$. Let $x \in I_1$ and observe that

$$\begin{aligned} 1-\overline{\mathscr{R}}_\kappa(-x) &= F_{\overline{\mathfrak{N}}}(\rho(1/2+x)) - F_{\overline{\mathfrak{N}}}(\rho(-1/2+x)) \\ &= \left[1-F_{\overline{\mathfrak{N}}}(-\rho(1/2+x))\right] - \left[1-F_{\overline{\mathfrak{N}}}(-\rho(-1/2+x))\right] \\ &= \left[1-F_{\overline{\mathfrak{N}}}(\rho(-1/2-x))\right] - \left[1-F_{\overline{\mathfrak{N}}}(\rho(1/2-x))\right] \\ &= F_{\overline{\mathfrak{N}}}(\rho(1/2-x)) - F_{\overline{\mathfrak{N}}}(\rho(-1/2-x)) \\ &= 1-\overline{\mathscr{R}}_\kappa(x) \end{aligned} \tag{46}$$

Claim 2: Observe that if $f_{\overline{\mathfrak{N}}}$ is even, then $F_{\overline{\mathfrak{N}}}(0) = 1/2$ and

$$\overline{\mathscr{R}}_\kappa(1/2) = 1 - \left[F_{\overline{\mathfrak{N}}}(0) - F_{\overline{\mathfrak{N}}}(-\rho)\right] = 1/2 + F_{\overline{\mathfrak{N}}}(-\rho) \ge 1/2 \tag{47}$$

Claim 3: Suppose $x > 0$ and observe that

$$\begin{aligned} \frac{d\overline{\mathscr{R}}_\kappa(x)}{dx} &= \rho f_{\overline{\mathfrak{N}}}(\rho(1/2-x)) - \rho f_{\overline{\mathfrak{N}}}(\rho(-1/2-x)) \\ &= \rho[f_{\overline{\mathfrak{N}}}(\rho(1/2-x)) - f_{\overline{\mathfrak{N}}}(\rho(1/2+x))] > 0 \end{aligned} \tag{48}$$

where we used the symmetry of $f_{\overline{\mathfrak{N}}}$ to replace $f_{\overline{\mathfrak{N}}}(\rho(-1/2-x))$ with $f_{\overline{\mathfrak{N}}}(\rho(1/2+x))$ and the fact that $1/2-x<1/2+x$ if $x>0$.

Claim 4: Derivate $\overline{\mathscr{R}}_{\kappa}(x)$ with respect to κ to obtain

$$\frac{d\overline{\mathscr{R}}_{\kappa}(x)}{d\kappa}=\left(\frac{1}{2}-x\right)f_{\overline{\mathfrak{N}}}\left(\kappa\left(\frac{1}{2}-x\right)\right)+\left(\frac{1}{2}+x\right)f_{\overline{\mathfrak{N}}}\left(\kappa\left(-\frac{1}{2}-x\right)\right) \tag{49}$$

If $|x|<1/2$, (49) is non negative; therefore, $\overline{\mathscr{R}}_{\kappa}(x)$ is a monotone increasing function of κ. □