Secure Sketch Metamorphosis: Tight Unified Bounds

Jeroen Delvaux[1,2], Dawu Gu[2], Ingrid Verbauwhede[1], Matthias Hiller[3] and Meng-Day (Mandel) Yu[4,1,5]

[1] ESAT/COSIC and iMinds, KU Leuven,
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium
{jeroen.delvaux, ingrid.verbauwhede}@esat.kuleuven.be
[2] CSE/LoCCS, Shanghai Jiao Tong University,
800 Dongchuan Road, Shanghai 200240, China
dwgu@sjtu.edu.cn
[3] Institute for Security in Information Technology,
Technische Universität München, Germany
matthias.hiller@tum.de
[4] Verayo Inc., USA
myu@verayo.com
[5] CSAIL, MIT, USA

Abstract. A noisy non-uniformly distributed secret often needs to be transformed into a stable high-entropy key. Biometric systems and *physically unclonable functions* (PUFs) exemplify the need for this conversion. *Secure sketches* are a useful tool hereby as they alleviate the noisiness while keeping the corresponding min-entropy loss to a minimum. The novelty of our work is twofold. First, seven secure sketch constructions, all based on a binary $[n, k, d]$ block code, are proven to be largely interchangeable. Despite having different looks and properties, all exhibit the same min-entropy loss, when fed with the same probability distribution. Second, for PUF-induced distributions with practical relevance, we derive new unified bounds on the min-entropy loss, considerably tighter than the more general well-known $(n - k)$ bound. Our bounds allow for an efficient evaluation and are hence suitable for reducing the implementation footprint of the sketch. This is beneficial for resource-constrained devices in particular.

Keywords: secure sketch, fuzzy extractor, min-entropy, physically unclonable functions, biometrics, coding theory

1 Introduction

Cryptography relies on reproducible uniformly distributed secret keys. Unfortunately, harvesting entropy from physical entities does not immediately fit within this framework. Biometric sensors and *physically unclonable functions* (PUFs) harvest from a human and an integrated circuit respectively. In both cases, measurements are corrupted by noise and non-uniformities are bound to occur. A

secure sketch, as part of a *fuzzy extractor* [9], provides a mechanism to convert such data into a high-quality key. This while providing *information-theoretic security*, i.e., irrespective of the computational power of an attacker.

1.1 Contribution

We consider seven secure sketch constructions: the syndrome method of Bennett et al. [3], the three code-offset variants of Juels et al. [14], Dodis et al. [9] and Tuyls et al. [22], the systematic methods of Yu [26] and Kang et al. [15] and finally the multi-code method of Ahlswede et al. [1]. The novelty of our work is twofold:

- First, we prove that all seven sketches have an identical min-entropy loss. At least, given that the underlying $[n, k, d]$ binary block code and ingoing probability distribution are the same. We stress that no constraints are imposed on the probability distribution. Therefore, our equivalencies reach considerably further than related work in [24, 12], covering fewer methods and establishing the link for uniformly distributed inputs only.
- Second, we derive new unified bounds on the min-entropy loss for PUF-induced distributions with practical relevance. Our bounds are considerably tighter than the well-known $(n-k)$ formula, hereby improving the implementation efficiency of PUF-based key generators. It is important to note that a variety of commonly used codes is covered, regardless of their algebraic complexity. Furthermore, a large variety of distributions could be supported. Therefore, our scope reaches considerable further than related work in [8, 18], focussing on simple repetition codes and biased distributions only. As in the latter works, our bounds are easy-to-evaluate and able to support large codes.

1.2 Organization

The remainder of this manuscript is organized as follows. Section 2 introduces notation and preliminaries. Section 3 revisits all seven secure sketch constructions. In Section 4, we prove the min-entropy loss equivalencies. In Section 5, we derive new unified bounds on the min-entropy loss, which are valid for all seven equivalent sketches. Section 6 concludes the work.

2 Preliminaries

2.1 Notation

Binary vectors are denoted with a bold lowercase character, e.g., $\boldsymbol{x}$. All vectors are row vectors. All-zeros and all-ones vectors are denoted with $\mathbf{0}$ and $\mathbf{1}$ respectively. Binary matrices are denoted with a bold uppercase character, e.g., $\boldsymbol{H}$. A random variable and its corresponding set of outcomes are denoted with an

uppercase *italic* and calligraphic character respectively, e.g., X and $\mathcal{X}$. Variable assignment is denoted with an arrow, e.g., $\boldsymbol{x} \leftarrow X$. Custom-defined procedure names are printed in a sans-serif font, e.g., Hamming weight $\mathsf{HW}(\boldsymbol{x})$ and Hamming distance $\mathsf{HD}(\boldsymbol{x}, \widetilde{\boldsymbol{x}})$. The probability of an event A is denoted as $\mathbb{P}(A)$. The expected value of a function $g(X)$ of random variable X is denoted as $\mathbb{E}_{x \leftarrow X}[g(X)]$.

2.2 Min-Entropy Definitions

The *min-entropy* of a random variable X is as defined in Equation (1). Consider now a pair of possibly correlated random variables: X and I. The *conditional min-entropy* [9] of X given I is as defined in Equation (2). Terms with $\mathbb{P}(I = i) = 0$ are evaluated as 0. Both definitions quantify the probability that an attacker guesses $x \leftarrow X$ first time right, on a logarithmic scale. Consider three possibly correlated random variables: X, I_1 and I_2. It was proven in [9] that Equation (3) holds. We stress that min-entropy is a more conservative notion than Shannon entropy and therefore often preferred within cryptology.

$$\mathbb{H}_\infty(X) = -\log_2\big(\max_{x \in \mathcal{X}} \mathbb{P}(X = x)\big). \tag{1}$$

$$\widetilde{\mathbb{H}}_\infty(X|I) = -\log_2\Big(\mathbb{E}_{i \leftarrow I}\big[\max_{x \in \mathcal{X}} \mathbb{P}((X = x)|(I = i))\big]\Big). \tag{2}$$

$$\widetilde{\mathbb{H}}_\infty(X|(I_1, I_2)) \geq \widetilde{\mathbb{H}}_\infty(X|I_1) - \log_2(|\mathcal{I}_2|). \tag{3}$$

2.3 Secure Sketches and Fuzzy Extractors

Consider a metric space $\mathcal{X}$ with distance function dist. An attacker knows the probability distribution of $x \leftarrow X$ and might be given some additional information $i \leftarrow I$. Variable I models external information that may be available about x regardless of the sketching procedure. For instance, X might capture a biometric with respect to the whole population and I could reshape the statistics according to age, gender, ethnicity, etc. For PUFs, I is considerably less relevant and typically omitted. Consider a noisy version $\widetilde{x}$ of sample x. An *average-case secure sketch* [9] is a pair of efficient and possibly randomized procedures: the sketching procedure $\boldsymbol{p} \leftarrow \mathsf{SSGen}(x)$, with helper data $\boldsymbol{p} \in \mathcal{P}$, and the recovery procedure $y \leftarrow \mathsf{SSRep}(\widetilde{x}, \boldsymbol{p})$, with $y \in \mathcal{Y}$. There are two defining properties:

- *Correctness.* If $\mathsf{dist}(x, \widetilde{x}) \leq t$, correctness of reconstruction is guaranteed, i.e., $\mathsf{SSRep}(x, \boldsymbol{p}) = \mathsf{SSRep}(\widetilde{x}, \boldsymbol{p})$. If $\mathsf{dist}(x, \widetilde{x}) > t$, there is no guarantee whatsoever.
- *Security.* For a certain lower-bound on the ingoing min-entropy, i.e., $\widetilde{\mathbb{H}}_\infty(X|I) \gtrapprox h_{in}$, there is a corresponding lower-bound on the residual min-entropy, i.e., $\widetilde{\mathbb{H}}_\infty(Y|(P, I)) \geq h_{out}$. Often, but not necessarily, this condition can be satisfied regardless of h_{in}. Or stated otherwise, there is a certain upper bound on the min-entropy loss $\Delta\mathbb{H}_\infty = \widetilde{\mathbb{H}}_\infty(X|I) - \widetilde{\mathbb{H}}_\infty(Y|(P, I))$.

A slightly modified notion brings us to the *average-case fuzzy extractor* [9]. Output $\boldsymbol{k} \in \mathcal{K}$ is then required to be nearly-uniform, given observations $\boldsymbol{p} \leftarrow P$ and $i \leftarrow I$, and is therefore suitable as a secret key. A more formal definition is given in Appendix A. Although secure sketches are our primary topic of interest, results automatically extend to fuzzy extractors. Simply because there are standard methods for crafting a fuzzy extractor from a secure sketch, as detailed again in Appendix A.

If we omit I, former constructions would reduce to a *secure sketch* and *fuzzy extractor* [9] respectively, i.e., without the *average-case*. In practice though, it is a frequent habit to omit the adjective *average-case* by default, even though it does apply. That's because many constructions exhibit the average-case property, e.g., all these in [9] as well as this work. Also, we generalized the original secure sketch definition so that the constraint $x \leftarrow \mathsf{SSRep}(x, \boldsymbol{p})$ does not apply anymore. As such, the prior notion of *fuzzy commitment* [14] is supported as well. Hereby, we commit to a secret value y by binding it to x. One may decommit given an $\widetilde{x}$ which is sufficiently close to x. Constructions which return a substring of $x = \boldsymbol{x}$, e.g., [15], are supported too. As a side note, the fuzzy extractor definition offers intrinsic support for both cases, without any modifications from our part.

2.4 Coding Theory

A *binary code* $\mathcal{C}$ is a bijection from a message space $\mathcal{M}$ to a codeword space $\mathcal{W} \subseteq \{0,1\}^{1\times n}$. The minimum distance d is the minimum number of bits in which any two distinct codewords differ. A procedure $\boldsymbol{w} \leftarrow \mathsf{Encode}(\boldsymbol{m})$ maps a message $\boldsymbol{m} \in \mathcal{M}$ to a codeword $\boldsymbol{w} \in \mathcal{W}$. A procedure $\widehat{\boldsymbol{w}} \leftarrow \mathsf{Correct}(\widetilde{\boldsymbol{w}})$ corrects up to $t = \lfloor \frac{d-1}{2} \rfloor$ errors for any noise-corrupted codeword $\widetilde{\boldsymbol{w}} = \boldsymbol{w} \oplus \boldsymbol{e}$, with $\mathsf{HW}(\boldsymbol{e}) \leq t$. An extended procedure $\widehat{\boldsymbol{m}} \leftarrow \mathsf{Decode}(\widetilde{\boldsymbol{w}})$ returns the corresponding message instead. Equation (4) expresses the Hamming bound [16]. The equality holds for *perfect codes* only, implicating that any vector in $\{0,1\}^{1\times n}$ is within distance t of a codeword. All other codes are subject to the inequality. A code is *optimal* if it has a maximum d for a given n and $|\mathcal{M}|$. A perfect code is optimal always, but the reverse is not necessarily true.

$$\sum_{i=0}^{t} \binom{n}{i} |\mathcal{M}| \leq 2^n. \tag{4}$$

A binary $[n, k, d]$ *block code* $\mathcal{C}$ restricts the message length $k = \log_2(|\mathcal{M}|)$ to an integer. For a linear block code, any linear combination of codewords is again a codeword. A $k \times n$ *generator matrix* $\boldsymbol{G}$, having full rank, can then implement the encoding procedure, i.e., $\boldsymbol{w} = \boldsymbol{m} \cdot \boldsymbol{G}$. A generator matrix is in *standard form* if $\boldsymbol{G} = (\boldsymbol{I}_k \| \boldsymbol{A})$. I.e., the first k bits of a codeword equal the message, followed by $n - k$ redundancy bits. A *parity check matrix* $\boldsymbol{H}$, with dimensions $(n - k) \times n$, determines the so-called *syndrome* $\boldsymbol{s} = \widetilde{\boldsymbol{w}} \cdot \boldsymbol{H}^T$. The syndrome captures all the information necessary for decoding $\widetilde{\boldsymbol{w}}$. For each codeword $\boldsymbol{w}$, the following holds: $\boldsymbol{0} = \boldsymbol{w} \cdot \boldsymbol{H}^T$. Therefore, the syndrome can be rewritten as $\boldsymbol{s} = \boldsymbol{e} \cdot \boldsymbol{H}^T$. Generator and parity check matrices can be derived from each other. E.g., for a generator

matrix in standard form, $\boldsymbol{H} = (\boldsymbol{A}^T \| \boldsymbol{I}_{n-k})$. The minimum distance d of a linear code equals the minimum Hamming weight of its nonzero codewords. A linear code $\mathcal{C}$ is *cyclic* if every circular shift of a codeword is again a codeword belonging to $\mathcal{C}$.

For any $\boldsymbol{\tau} \in \{0,1\}^{1\times n}$ and linear code $\mathcal{C}$, the set $\{\boldsymbol{\tau} \oplus \boldsymbol{w} : \boldsymbol{w} \in \mathcal{W}\}$ is referred to as a *coset*. Two cosets are either disjoint or coincide. Therefore, the vector space $\{0,1\}^{1\times n}$ is fully covered by 2^{n-k} cosets, referred to as the *standard array*. The minimum weight vector $\boldsymbol{\epsilon}$ in a coset is called the *coset leader*. In case of conflict, i.e., a common minimum $\mathsf{HW}(\boldsymbol{\epsilon}) > t$, an arbitrary leader can be selected. There is a one-to-one correspondence between cosets and syndromes.

3 Secure Sketch Constructions - Revisited

We assume x to be a binary vector, i.e., $x = \boldsymbol{x}$. For PUFs, this is generally speaking the case; for biometrics, this might involve explicit quantization [21]. Figure 1 represents all seven secure sketch constructions, instantiated with a binary code $\mathcal{C}$. An eight secure sketch, Davida et al. [7], is discussed in Appendix B as its performance is considerably lower. Distance function dist is instantiated with Hamming distance HD. For many codes in literature (BCH, Hamming, repetition, etc.), there are efficient decoding algorithms which guarantee correctness if $\mathsf{HD}(\boldsymbol{x}, \widetilde{\boldsymbol{x}}) \leq t$ [16]. We stress that the min-entropy loss $\Delta\mathbb{H}_\infty$ does not depend on the decoding method, simply because the helper data is not affected.

3.1 Syndrome Method of Bennett et al.

The syndrome method of Bennett et al. [3] is represented by Figure 1(a). Although initially proposed as part of a *quantum oblivious transfer* protocol, it maps quite easily to the secure sketch framework of Dodis et al. [9]. The method requires a linear code $\mathcal{C}$, given the use of a parity check matrix $\boldsymbol{H}$. The well-known $(n-k)$ upper bound on the min-entropy loss $\Delta\mathbb{H}_\infty$ holds, as proven by Dodis et al. [9]. This is a trivial consequence from Equation (3), given that the helper data $\boldsymbol{p}$ is limited to $(n-k)$ bits.

3.2 Code-Offset Methods of Juels et al., Dodis et al. and Tuyls et al.

The code-offset method of Juels et al. [14] is represented by Figure 1(b). The code $\mathcal{C}$ is not necessarily linear. Even more, it is not required be a block code either. Entropy loss can be understood as a *one-time pad* imperfection. Sketch input $\boldsymbol{x}$ is masked with a random codeword $\boldsymbol{w}$, i.e., an inherent entropy deficiency: $\mathbb{H}_\infty(W) = \log_2(|\mathcal{M}|) < n$. Figure 1(c) represents a modification where Rep returns sketch input $\boldsymbol{x}$ rather than codeword $\boldsymbol{w}$, as proposed by Dodis et al. [9]. For the latter, it was proven that the $(n-k)$ upper bound on the min-entropy loss $\Delta\mathbb{H}_\infty$ holds, given a block code. The proof is more complicated as

$\boldsymbol{p} \leftarrow \mathsf{SSGen}(\boldsymbol{x})$	$\widehat{\boldsymbol{y}} \leftarrow \mathsf{SSRep}(\widetilde{\boldsymbol{x}}, \boldsymbol{p})$	
$\boldsymbol{p} \leftarrow \boldsymbol{x} \cdot \boldsymbol{H}^T$	$\boldsymbol{s} \leftarrow \widetilde{\boldsymbol{x}} \cdot \boldsymbol{H}^T \oplus \boldsymbol{p} = \boldsymbol{e} \cdot \boldsymbol{H}^T$ Determine $\widehat{\boldsymbol{e}}$ $\widehat{\boldsymbol{y}} = \widehat{\boldsymbol{x}} \leftarrow \widetilde{\boldsymbol{x}} \oplus \widehat{\boldsymbol{e}}$	(a) Syndrome method of Bennett et al. [3].
Random $\boldsymbol{w} \in \mathcal{C}$ $\boldsymbol{p} \leftarrow \boldsymbol{x} \oplus \boldsymbol{w}$	$\widetilde{\boldsymbol{w}} \leftarrow \widetilde{\boldsymbol{x}} \oplus \boldsymbol{p} = \boldsymbol{w} \oplus \boldsymbol{e}$ $\widehat{\boldsymbol{y}} = \widehat{\boldsymbol{w}} \leftarrow \mathsf{Correct}(\widetilde{\boldsymbol{w}})$	(b) Code-offset method of Juels et al. [14].
	$\widetilde{\boldsymbol{w}} \leftarrow \widetilde{\boldsymbol{x}} \oplus \boldsymbol{p} = \boldsymbol{w} \oplus \boldsymbol{e}$ $\widehat{\boldsymbol{y}} = \widehat{\boldsymbol{x}} \leftarrow \boldsymbol{p} \oplus \mathsf{Correct}(\widetilde{\boldsymbol{w}})$	(c) Code-offset method of Dodis et al. [9].
	$\widetilde{\boldsymbol{w}} \leftarrow \widetilde{\boldsymbol{x}} \oplus \boldsymbol{p} = \boldsymbol{w} \oplus \boldsymbol{e}$ $\widehat{\boldsymbol{y}} = \widehat{\boldsymbol{m}} \leftarrow \mathsf{Decode}(\widetilde{\boldsymbol{w}})$	(d) Code-offset method of Tuyls et al. [22].
$\boldsymbol{p} \leftarrow \boldsymbol{x}(1:k) \cdot \boldsymbol{A}$ $\oplus\ \boldsymbol{x}(k+1:n)$	$\widehat{\boldsymbol{w}} \leftarrow \mathsf{Correct}(\widetilde{\boldsymbol{x}} \oplus (\boldsymbol{0} \| \boldsymbol{p}))$ $\widehat{\boldsymbol{y}} = \widehat{\boldsymbol{x}} \leftarrow \widehat{\boldsymbol{w}} \oplus (\boldsymbol{0} \| \boldsymbol{p})$	(e) Systematic method of Yu [26].
	$\widehat{\boldsymbol{y}} = \widehat{\boldsymbol{x}}(1:k) \leftarrow \mathsf{Decode}(\widetilde{\boldsymbol{x}}$ $\oplus(\boldsymbol{0} \| \boldsymbol{p}))$	(f) Systematic method of Kang et al. [15].
$\boldsymbol{p} \leftarrow j$ so that $\boldsymbol{x} \in \mathcal{C}_j$	$\widehat{\boldsymbol{y}} = \widehat{\boldsymbol{m}} \leftarrow \mathsf{Decode}_{\mathcal{C}_j}(\widetilde{\boldsymbol{x}})$	(g) Multi-code method of Ahlswede et al. [1].

Fig. 1. Seven secure sketch constructions, all having an n-bit input $\boldsymbol{x}$. Correctness is guaranteed, given a noisy version $\widetilde{\boldsymbol{x}}$ with $\mathsf{HD}(\boldsymbol{x}, \widetilde{\boldsymbol{x}}) \leq t$.

for the syndrome construction and hence not repeated here. Figure 1(d) represents another minor modification where Rep returns message $\boldsymbol{m}$, as suggested by Tuyls et al. [22]. This necessitates an implementation of Decode rather than Correct.

3.3 Systematic Methods of Yu and Kang et al.

The method of Yu [26] is represented by Figure 1(e). It requires a linear code $\mathcal{C}$ with the generator matrix in standard form, i.e., $\boldsymbol{G} = (\boldsymbol{I}_k \| \boldsymbol{A})$. We observe that $\Delta\mathbb{H}_\infty \leq (n-k)$ holds due to Equation (3), given that helper data $\boldsymbol{p}$ is limited to $(n-k)$ bits. Figure 1(f) represents a slightly modified method where Rep returns $\boldsymbol{x}(1:k)$ rather than $\boldsymbol{x}$. This was first proposed by Kang et al. in [15] and independently also by Hiller et al. in [12].

3.4 Multi-Code Method of Ahlswede et al.

The method of Ahlswede et al. [1] is represented by Figure 1(g). Although initially proposed for secret key transport with *correlated sources*, it maps quite easily to our framework of interest, as observed by Hiller et al. [12]. A distinguishing feature is the use of multiple codes $\mathcal{C}_j$, covering mutually disjoint sets of codewords. We restrict our attention to $[n, k, d]$ block codes with $j \in [0, 2^{n-k}-1]$. Every $\boldsymbol{x} \in \mathcal{X}$ then coincides with exactly one codeword, guaranteeing correctness. Furthermore, $\Delta\mathbb{H}_\infty \leq (n-k)$ holds due to Equation (3), given that helper data $\boldsymbol{p} = j$ is limited to $(n-k)$ bits. In [12], Hiller et al. proposed an efficient implementation where all codes are derived from a single parent code $\mathcal{C}_0$. In particular, $\mathcal{C}_0$ is a linear code in standard form, i.e., $\boldsymbol{G} = (\boldsymbol{I}_k \| \boldsymbol{A})$, and all other codes are cosets: $\mathcal{C}_j = \{\boldsymbol{w} \oplus (\boldsymbol{0} \| \boldsymbol{p}) \mid \boldsymbol{w} \in \mathcal{C}_0\}$. This turns out to be fully equivalent with the method of Kang et al. in Figure 1(f), i.e., helper data $\boldsymbol{p}$ and reconstructed output $\boldsymbol{y}$ are identical.

4 Metamorphosis: Entropy Loss Equivalencies

For several secure sketch constructions, the $(n-k)$ upper bound on the min-entropy loss was shown to be valid. In this Section, we prove that equivalencies reach considerably further. An overview is provided in Figure 2. All seven constructions exhibit an identical min-entropy loss. Or more precisely, all have the same residual min-entropy, given by Equation (5), as long as the ingoing distribution (X, I) and the code $\mathcal{C}$ are identical. Remember that terms with $\mathbb{P}((P = \boldsymbol{p}) \cap (I = i)) = 0$ are treated as 0. Also, we note that the equivalencies easily extend to Shannon entropy, as proven in Appendix C.

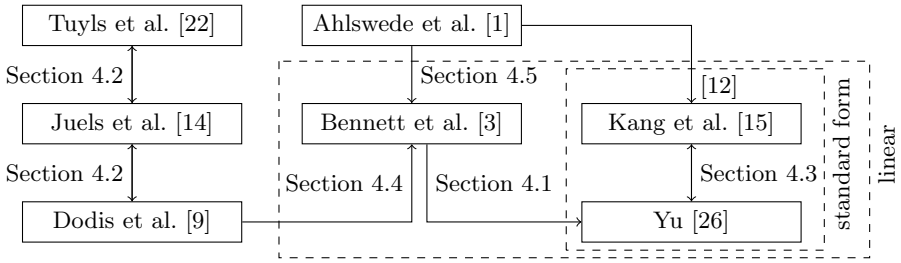


Fig. 2. Entropy loss equivalencies. Transitive relations apply when following the arrows. E.g., the schemes of Dodis et al. and Kang et al. are equivalent, given that both are instantiated with a linear code in standard form.

$$\widetilde{\mathbb{H}}_\infty(Y|(P,I)) = -\log_2\Big(\mathbb{E}_{(\boldsymbol{p},i)\leftarrow(P,I)}\big[\max_{\boldsymbol{y}\in\mathcal{Y}}\mathbb{P}((Y=\boldsymbol{y})|((P=\boldsymbol{p})\cap(I=i)))\big]\Big). \quad (5)$$

4.1 Bennett et al. versus Yu

The methods of Bennett et al. and Yu both reconstruct the sketch input, i.e., $\boldsymbol{y} = \boldsymbol{x}$. We are the first to observe though that the helper data is identical as well, as proven in Equation (6). Of course, this assumes a generator matrix in standard form, i.e., $\boldsymbol{G} = (\boldsymbol{I}_k \| \boldsymbol{A})$, given that Yu's method is restricted to this case.

$$\boldsymbol{p} = \boldsymbol{x} \cdot \boldsymbol{H}^T = \boldsymbol{x} \cdot \begin{pmatrix} \boldsymbol{A} \\ \boldsymbol{I}_{n-k} \end{pmatrix} = \boldsymbol{x}(1:k) \cdot \boldsymbol{A} \oplus \ \boldsymbol{x}(k+1:n). \tag{6}$$

4.2 Juels et al. versus Dodis et al. versus Tuyls et al.

All three code-offset methods produce the same helper data $\boldsymbol{p}$ but differ in their reconstructed output $\boldsymbol{y}$. Nevertheless, we argue that the residual min-entropy is identical. This follows from an underlying one-to-one correspondence, given by Equation (7). Encode comprehends a bijection between message space $\mathcal{M}$ and codeword space $\mathcal{W}$. Furthermore, for a given $\boldsymbol{p}$, there is a bijection between $\mathcal{W}$ and a reduced response space $\mathcal{X}' = \{\boldsymbol{p} \oplus \boldsymbol{w} \mid \boldsymbol{w} \in \mathcal{W}\} \subseteq \mathcal{X}$. Therefore, Equation (5) evaluates to the same value for all three methods. Note that $|\mathcal{M}| = |\mathcal{W}| = |\mathcal{X}'|$.

$$\begin{aligned} \forall(\boldsymbol{p}, i, \boldsymbol{m}) \in (\mathcal{P} \times \mathcal{I} \times \mathcal{M}), \mathbb{P}((M = \boldsymbol{m}) | ((P = \boldsymbol{p}) \cap (I = i))) \\ = \mathbb{P}((W = \mathsf{Encode}(\boldsymbol{m})) | ((P = \boldsymbol{p}) \cap (I = i))) \\ = \mathbb{P}((X = \mathsf{Encode}(\boldsymbol{m}) \oplus \boldsymbol{p}) | ((P = \boldsymbol{p}) \cap (I = i))). \end{aligned} \tag{7}$$

4.3 Yu versus Kang et al.

The methods of Yu and Kang et al. produce the same helper data $\boldsymbol{p}$ but reconstruct $\boldsymbol{y} = \boldsymbol{x}$ and $\boldsymbol{y} = \boldsymbol{x}(1:k)$ respectively. Nevertheless, Equation (8) indicates that the residual min-entropy is identical. The main insight is that $\boldsymbol{x}(1:k)$ and $\boldsymbol{p}$ fully determine $\boldsymbol{x}(k+1:n)$.

$$\begin{aligned} \forall(\boldsymbol{p}, i, \boldsymbol{x}) \in (\mathcal{P} \times \mathcal{I} \times \mathcal{X}), \mathbb{P}((X(1:k) = \boldsymbol{x}(1:k)) | ((P = \boldsymbol{p}) \cap (I = i))) \\ = \mathbb{P}((X = (\boldsymbol{x}(1:k) \| (\boldsymbol{x}(1:k) \cdot \boldsymbol{A} \oplus \boldsymbol{p})) | ((P = \boldsymbol{p}) \cap (I = i))). \end{aligned} \tag{8}$$

4.4 Bennett et al. versus Dodis et al.

The syndrome method of Bennett et al. and the code-offset method of Dodis et al. both reconstruct $\boldsymbol{y} = \boldsymbol{x}$. Furthermore, for both methods, helper data $\boldsymbol{p}$ reveals in which coset $\boldsymbol{x}$ resides. For the syndrome method, this is a trivial consequence from the one-to-one correspondence between cosets and syndromes. For the code-offset method, $\boldsymbol{p}$ comprehends a random element in the same coset as $\boldsymbol{x}$. Note that the code-offset method is being instantiated with a linear code,

given that the syndrome method is restricted to this case. Equation (5) can hence be rewritten as shown in Equation (9), with $\boldsymbol{\epsilon}$ the coset leader. Remember that I models external information about x, regardless of the sketching algorithm, so it remains invariant given our change of variable.

$$\widetilde{\mathbb{H}}_\infty(X|(P,I)) = -\log_2\Big(\mathbb{E}_{(\boldsymbol{\epsilon},i)\leftarrow(E,I)}\big[\max_{\boldsymbol{w}\in\mathcal{C}}\mathbb{P}((X=\boldsymbol{\epsilon}\oplus\boldsymbol{w})|((E=\boldsymbol{\epsilon})\cap(I=i)))\big]\Big). \tag{9}$$

4.5 Bennett et al. versus Ahlswede et al.

For the multi-code method of Ahlswede et al., we need to assume that all codes $\mathcal{C}_j$ are derived from a single parent code $\mathcal{C}_0$. As such, Hiller et al. [12] established an equivalence with the sketch of Kang et al., given $\mathcal{C}_0$ linear and in standard form. We consider a slightly more general case. In particular, a linear code $\mathcal{C}_0$ which is not necessarily in standard form, as required by the method of Bennett et al. as well. All child codes $\mathcal{C}_j$ are again formed as the cosets of $\mathcal{C}_0$. Therefore, helper data $\boldsymbol{p} = j$ still reveals in which coset $\boldsymbol{x}$ resides and Equation (9) holds once again. The one-to-one correspondence of output $\boldsymbol{y}$ in Equation (10) finalizes our proof.

$$\begin{aligned}\forall(\boldsymbol{p},i,\boldsymbol{x})\in(\mathcal{P}\times\mathcal{I}\times\mathcal{X}),\mathbb{P}((X=\boldsymbol{x})|((P=\boldsymbol{p})\cap(I=i)))\\ =\mathbb{P}((M=\mathsf{Decode}_{\mathcal{C}_{\boldsymbol{p}}}(\boldsymbol{x}))|((P=\boldsymbol{p})\cap(I=i))).\end{aligned} \tag{10}$$

4.6 Generalization: Concatenated Codes in Parallel

The implementation footprint of $\mathsf{Correct}/\mathsf{Decode}$ imposes upper bounds on code size parameters $[n,k,d]$. Therefore, in order to generate a key of sufficient length, z instances of a smaller code $[n_1,k_1,d_1]$ are typically applied in parallel. Furthermore, for high error rates in particular, concatenated codes are often used [4]. As a generalization, we consider z instances of $[n_2,k_2,d_2]\circ[n_1,k_1,d_1]$, with n_1 an integer multiple of k_2. One could think of these as a single *umbrella* block code with $n = z\cdot n_2\cdot\frac{n_1}{k_1}$ and $k = z\cdot k_1$. Therefore, prior equivalencies still apply.

4.7 Discussion

We adopt the perspective of an interested system provider, aiming to select a sketch. Our newly proven equivalencies considerably simplify the selection procedure. In particular, entropy loss is not a distinguisher. Table 1 lists various other factors which might affect decision making.

In general, a secure sketch needs to be used as part of a fuzzy extractor in order to obtain a uniformly distributed key $\boldsymbol{k}$. This often boils down to an additional hashing step, i.e., $\boldsymbol{k}\leftarrow\mathsf{Hash}(\boldsymbol{y})$. However, there are exceptions where the sketch by itself might be sufficient [12]. Consider an n-bit input X which

	Bennett et al. [3]	Juels et al. [14]	Dodis et al. [9]	Tuyls et al. [22]	Yu [26]	Kang et al. [15]	Ahlswede et al. [1]
Ingoing bits	n	n	n	n	n	n	n
Outgoing bits	n	n	n	k	n	k	k
Helper bits	$n-k$	n	n	n	$n-k$	$n-k$	$n-k$
Commitment	no	yes	no	yes	no	no	no
Gen deterministic	yes	no	no	no	yes	yes	yes
Entropy loss	Equivalent. Bounded by $\Delta\mathbb{H}_\infty \leq n-k$.						
Code requirements	L	/	/	/	L S	L S	

Table 1. Secure sketch comparison for an $[n, k, d]$ block code. For the code requirements, 'L' denotes linear and 'S' denotes systematic.

is (nearly) uniformly distributed and given I absent, as might be the case for certain PUFs. Sketches which a k-bit output then maintain this (nearly) uniform distribution, according to the $(n-k)$ bound. In particular, if $\mathbb{H}_\infty(X) = n - \epsilon$, then $\widetilde{\mathbb{H}}_\infty(Y|P) \geq k - \epsilon$, given a presumably small $\epsilon \geq 0$.

Furthermore, we consider two conflicting properties regarding the randomness of Gen. For a randomized Gen, fuzzy commitment could be considered as an advantage, as it allows for more control on the key. As an example, one could easily replace a malfunctioning PUF device without having to change the key. On the other hand, a deterministic Gen eliminates the need for high-quality random numbers. This is interesting if Gen is implemented on-chip in particular.

As another distinguisher, helper data size could be considered. Obviously, $(n-k)$ bits is preferred above n bits. Finally, a code needs to be selected. As mentioned in [9], dense codes have the benefit. In the ideal case, this would be a perfect/optimal code. For the same min-entropy loss, i.e., bound $(n-k)$, more errors, i.e., t, can be corrected then.

5 Tight Unified Bounds

Currently, secure sketch implementations rely on the $(n-k)$ upper bound on the min-entropy loss, e.g., [19]. Unfortunately, this leads to an overly conservative design when instantiating security parameters accordingly. Another problem is that the ingoing min-entropy $\mathbb{H}_\infty(X)$ cannot be determined exhaustively. As n ranges from hundreds to thousands of bits in order to generate a key of sufficient length, e.g., 128 bit, one cannot simply measure the probability of occurrence of the most likely value $\boldsymbol{x} \in \mathcal{X}$. For PUFs in particular, one would have to manufacture and read-out an infeasible number of devices for this purpose. E.g., $\gg 2^{1000}$, for $n = 1000$, although depending on the distribution as well as the

confidence level. Therefore, theoretical models are unavoidable, allowing to estimate min-entropy based on a limited number of samples. So in summary, the pessimistic $(n-k)$ bound is applied to a good estimate of $\mathbb{H}_\infty(X)$ at best.

We propose a more efficient substitute for the $(n-k)$ bound, targeting PUF-based key generation in particular. A prominent category of PUFs consists of an array of identically designed cells, each producing a single bit, or occasionally a few bits. This includes memory-based designs, such as the SRAM PUF [13], as well as the coating PUF [23] and a subset of the large number of ring oscillator-based designs, e.g., [25]. The most prominent entropy-degrading effects for such PUFs are bias and spatial correlations. Bias comprehends an imbalance between the number of zeros and ones. Spatial correlations implicate that neighboring cells might influence each other. Former effects might be incorporated in a model for distribution X, allowing to estimate $\mathbb{H}_\infty(X)$ as such.

We develop a graphical framework that produces tight bounds on $\widetilde{\mathbb{H}}_\infty(Y|P)$ for PUF-induced distributions. The critical *first-order* effects of bias and spatial correlations are captured. Both lower and upper bounds are supported. The lower bounds are of primary interest for a conservative system provider, entertaining the worst-case scenario. We considerably improve upon the $(n-k)$ bound, i.e., the leftmost inequality in Equation (11). We also improve upon the rather trivial upper bounds which comprehend the rightmost inequality in Equation (11). The validity of the $\log_2(|\mathcal{M}|)$ bound may be argued from the code-offset method of Tuyls et al., having a number of outcomes limited to $|\mathcal{Y}| = |\mathcal{M}|$. The equivalencies in Section 4 subsequently impose this bound on other sketches.

$$\underbrace{\max(\mathbb{H}_\infty(X) - (n-k), 0)}_{\text{worst-case}} \leq \widetilde{\mathbb{H}}_\infty(Y|P) \leq \underbrace{\min(\log_2(|\mathcal{M}|), \mathbb{H}_\infty(X))}_{\text{best-case}}. \tag{11}$$

Our lower and upper bounds combined define a relatively narrow interval in which the exact value of $\widetilde{\mathbb{H}}_\infty(Y|P)$ is enclosed. We considerably extend related work in [8, 18] as follows. First, we cover a variety of codes, regardless of their algebraic complexity. Prior work focussed on repetition codes only. Although frequently used as the inner code of a concatenated code, full-fledged key generators typically rely on non-trivial codes, e.g., BCH codes [16, 19]. Second, our techniques may be applied to a variety of distributions, while prior work covered biased distributions only. Note that our bounds remain easy-to-evaluate and able to handle large codes.

5.1 Distributions

Our work is generic in the sense that a large variety of distributions X could be covered. We only require that $\mathcal{X} = \{0,1\}^{1\times n}$ can be partitioned in subsets φ_j, with $j \in [1, J]$, so that all elements of φ_j have the same probability of occurrence q_j. Formally, $\mathbb{P}(X = \boldsymbol{x}) = q_j$ if and only if $\boldsymbol{x} \in \varphi_j$. These probabilities are strictly monotonically decreasing, i.e., $q_1 > q_2 > \ldots > q_J$. Occasionally, $q_J = 0$. The ingoing min-entropy is easily computed as $\mathbb{H}_\infty(X) = -\log_2(q_1)$. We determine

bounds on $\widetilde{\mathbb{H}}_\infty(Y|P)$. The runtime of the corresponding algorithms is roughly proportional to J. The crucial observation is that even a very small J might suffice to capture realistic PUF models. Below, we describe a parameterized distribution R for both biased and spatially correlated PUFs.

- *Biased distribution.* We assume bits to be independent and identically distributed so that $\mathbb{P}(X(i) = 1) = b$, with $i \in [1, n]$ and a real-valued $b \in [0, 1]$. For $b = \frac{1}{2}$, this boils down to a uniform distribution. As a side note, this model comprehends a very popular abstraction in PUF literature. The min-entropy loss of various other helper data methods has been analyzed as such, e.g., *IBS* [27] and *soft-decision decoding* [17, 8]. Therefore, our results enable adequate comparison with related methods, all using a common baseline distribution.
- *Correlated distribution.* We assume bits to be distributed so that $\mathbb{P}(X(i) = X(i+1)) = c$, with $i \in [1, n-1]$ and a real-valued $c \in [0, 1]$. There is no bias. For $c = \frac{1}{2}$, this boils down to a uniform distribution. Although spatial correlations are generally acknowledged to be an issue, these are usually ignored in theoretical work due to their complexity. We hope that our results may help turn the tide on this.

Figure 3 specifies the subsets φ_j for both distributions. For the biased distribution, we partition according to $\mathsf{HW}(\boldsymbol{x})$. Essentially, this boils down to a binomial distribution with $j-1$ successes for n Bernoulli trials, each having success probability $b_\star = \min(b, 1-b)$. For the correlated distribution, we partition according to $\mathsf{HD}(\boldsymbol{x}(1 : n-1), \boldsymbol{x}(2 : n))$, i.e., the number of transitions. Inputs in subset φ_j exhibit $j-1$ transitions and obey either one out of two forms, i.e., $\boldsymbol{x} = (\mathbf{0}\|\mathbf{1}\|\mathbf{0}\|\ldots)$ and $\boldsymbol{x} = (\mathbf{1}\|\mathbf{0}\|\mathbf{1}\|\ldots)$. A related observation is that if $\boldsymbol{x} \in \varphi_j$, then so is its ones' complement, i.e., $\overline{\boldsymbol{x}} \in \varphi_j$. This explains the factors 2 and $\frac{1}{2}$ everywhere. Set size $|\varphi_j|$ is further determined with *stars and bars* combinatorics [10]. In particular, we separate n indistinguishable stars into j distinguishable bins by adding $j-1$ out of $n-1$ bars.

j	$\lvert\varphi_j\rvert$	q_j
1	1	$(1-b_\star)^n$
2	n	$b_\star(1-b_\star)^{n-1}$
$\ldots$	$\ldots$	$\ldots$
j	$\binom{n}{j-1}$	$(b_\star)^{j-1}(1-b_\star)^{n-j+1}$
$\ldots$	$\ldots$	$\ldots$
n	n	$(b_\star)^{n-1}(1-b_\star)$
$n+1$	1	$(b_\star)^n$

j	$\lvert\varphi_j\rvert$	q_j
1	2	$\frac{1}{2}(1-c_\star)^{n-1}$
2	$2(n-1)$	$\frac{1}{2}c_\star(1-c_\star)^{n-2}$
$\ldots$	$\ldots$	$\ldots$
j	$2\binom{n-1}{j-1}$	$\frac{1}{2}(c_\star)^{j-1}(1-c_\star)^{n-j}$
$\ldots$	$\ldots$	$\ldots$
$n-1$	$2(n-1)$	$\frac{1}{2}(c_\star)^{n-2}(1-c_\star)$
n	2	$\frac{1}{2}(c_\star)^{n-1}$

Fig. 3. Subsets φ_j for a biased and correlated distribution X, left and right respectively. We define $b_\star = \min(b, 1-b)$ and $c_\star = \min(c, 1-c)$.

We treat the degenerate case $b = c = \frac{1}{2}$, i.e., a uniform distribution, separately. There is only one set then. Formally, $J = 1$, $|\varphi_1| = 2^n$ and $q_1 = 1/2^n$. As proven by Reyzin [20], the min-entropy loss of a secure sketch is maximal for a uniformly distributed input, making this a case of special interest.

5.2 Generic Bounds

Due to the newly proven equivalencies in Section 4, we can limit the analysis to a single sketch only. We opt for the code-offset construction of Dodis et al., as it does not impose restrictions on the code, linearity in particular. This maximizes the generality of our results. Equation (12) holds, given that a codeword is selected fully at random during enrollment.

$$\mathbb{P}((P = \boldsymbol{p})|(X = \boldsymbol{x})) = \begin{cases} 1/|\mathcal{M}|, & \text{if } \exists \boldsymbol{w} : \boldsymbol{p} = \boldsymbol{x} \oplus \boldsymbol{w} \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

Equation (13) resumes Equation (5), while omitting I. Also, we apply Bayes' rule and fill in Equation (12). The 0 case is resolved by switching variables for the max operator. A direct exhaustive evaluation of the resulting formula requires up to $2^n|\mathcal{M}|$ operations.

$$\begin{aligned} \widetilde{\mathbb{H}}_\infty(Y|P) &= -\log_2\left(\sum_{\boldsymbol{p}\in\mathcal{P}} \cancel{\mathbb{P}(P=\boldsymbol{p})} \max_{\boldsymbol{x}\in\mathcal{X}} \frac{\mathbb{P}(X=\boldsymbol{x})\mathbb{P}((P=\boldsymbol{p})|(X=\boldsymbol{x}))}{\cancel{\mathbb{P}(P=\boldsymbol{p})}}\right) \\ &= -\log_2\left(\frac{1}{|\mathcal{M}|}\sum_{\boldsymbol{p}\in\mathcal{P}} \max_{\boldsymbol{w}\in\mathcal{W}} \mathbb{P}(X=\boldsymbol{p}\oplus\boldsymbol{w})\right). \end{aligned} \tag{13}$$

For linear codes, the workload can be reduced substantially. With a similar derivation as before, we rewrite Equation (9) as shown in Equation (14). Up to 2^n operations suffice. Nevertheless, direct evaluation is only feasible for small codes. We stress that our bounds are able to handle large codes, as is typically the case for a practical key generator.

$$\widetilde{\mathbb{H}}_\infty(Y|P) = -\log_2\Big(\sum_{\boldsymbol{\epsilon}\in\mathcal{E}} \max_{\boldsymbol{w}\in\mathcal{W}} \mathbb{P}(X=\boldsymbol{\epsilon}\oplus\boldsymbol{w})\Big). \tag{14}$$

Equation (13) iterates over all $\boldsymbol{p}$'s and selects each time the most likely $\boldsymbol{x}$ which is within range, via the addition of a codeword $w \in \mathcal{W}$. We now reverse the roles, as shown in Figure 4. We iterate over all $\boldsymbol{x}$'s, from most likely to least likely, i.e., from φ_1 to φ_J. Within a certain φ_j, the order of the $\boldsymbol{x}$'s may be chosen arbitrarily. Subsequently, we assign $\boldsymbol{p}$'s to each $\boldsymbol{x}$, as represented by the black squares, until the set $\mathcal{P}$ of size 2^n is depleted. For each assigned $\boldsymbol{p}$, we assume that the corresponding $\boldsymbol{x}$ is the most likely vector, according to Equation (13). Let $s_j^{\boldsymbol{p}}$ denote the number of black squares assigned to set φ_j. The residual min-entropy is then easily computed as in Equation (15).

$$\widetilde{\mathbb{H}}_\infty(Y|P) = -\log_2\left(\frac{1}{|\mathcal{M}|}\sum_{j=1}^{J} s_j^{\boldsymbol{p}} q_j\right). \tag{15}$$

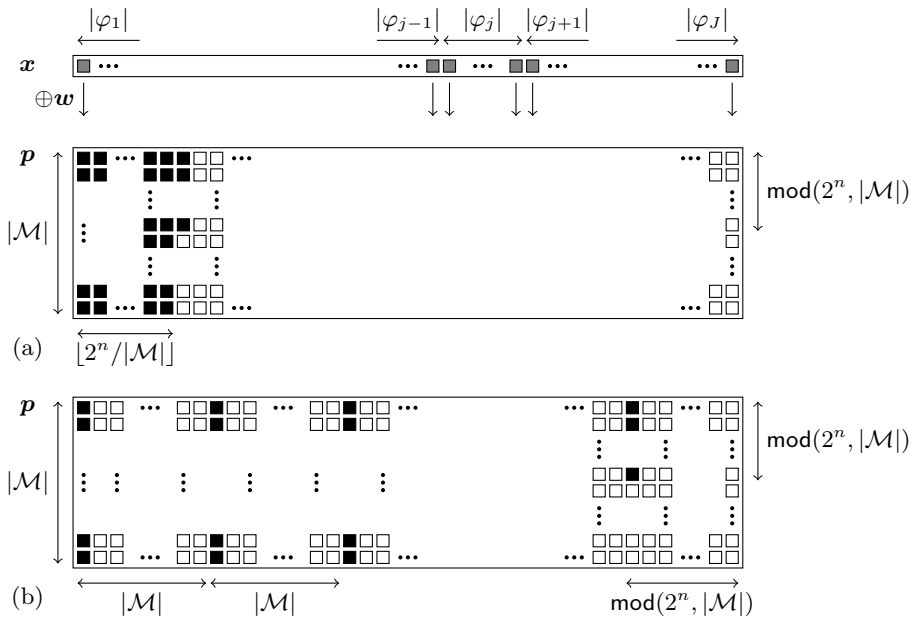


Fig. 4. Reversal of the roles in Equation (13). (a) A lower bound on $\widetilde{\mathbb{H}}_\infty(Y|P)$. (b) An upper bound on $\widetilde{\mathbb{H}}_\infty(Y|P)$. Black squares represent terms which contribute to $\widetilde{\mathbb{H}}_\infty(Y|P)$, one for each $\boldsymbol{p} \in \mathcal{P}$. White squares represent non-contributing terms, overruled by the max operator. In general, there are few black squares but many white squares, 2^n versus $(|\mathcal{M}|-1)2^n$ to be precise. For block codes, i.e., $|\mathcal{M}| = 2^k$, the last column of black squares is completely filled.

Both linear and non-linear codes are supported by former graphical representation. Nevertheless, we elaborate linear codes as a special case so as to improve the insights. Figure 5 swaps the order of iteration in Equation (14). Only one row suffices, i.e., each column of helper data vectors $\boldsymbol{p}$ in Figure 4 is condensed to a single square. Black and white squares are now assigned to cosets, as represented by their coset leaders $\boldsymbol{\epsilon}$. Let $s_j^{\boldsymbol{\epsilon}}$ denote the number of black squares assigned to set φ_j. The residual min-entropy is then easily computed as in Equation (16), hereby dropping denominator $|\mathcal{M}|$ compared to Equation (15), given that $s_j^{\boldsymbol{p}} = 2^k \cdot s_j^{\boldsymbol{\epsilon}}$.

$$\widetilde{\mathbb{H}}_\infty(Y|P) = -\log_2\left(\sum_{j=1}^{J} s_j^{\boldsymbol{\epsilon}} q_j\right). \tag{16}$$

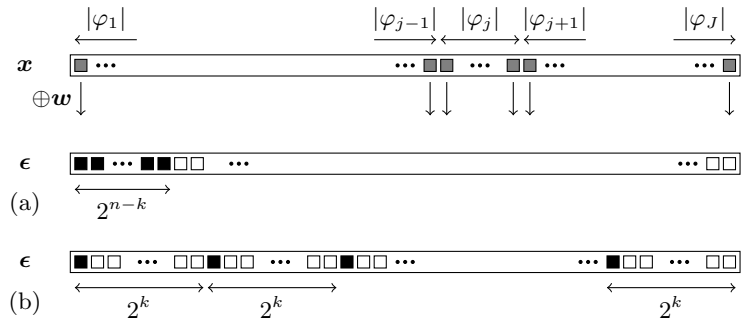


Fig. 5. Reversal of the roles in Equation (14), as applied to linear codes. (a) A lower bound on $\widetilde{\mathbb{H}}_\infty(Y|P)$. (b) An upper bound on $\widetilde{\mathbb{H}}_\infty(Y|P)$. Black squares represent terms which contribute to $\widetilde{\mathbb{H}}_\infty(Y|P)$, one for each $\boldsymbol{\epsilon} \in \mathcal{E}$. White squares represent non-contributing terms, overruled by the max operator.

In the worst-case scenario, the most likely $\boldsymbol{x}$'s all map to unique $\boldsymbol{p}$'s, without overlap, resulting in a lower bound on $\widetilde{\mathbb{H}}_\infty(Y|P)$. For a linear code, this would be the case if the first 2^{n-k} $\boldsymbol{x}$'s all belong to different cosets. In the best-case scenario, our sequence of $\boldsymbol{x}$'s exhibits maximum overlap in terms of $\boldsymbol{p}$, resulting in an upper bound on $\widetilde{\mathbb{H}}_\infty(Y|P)$. For a linear code, this would be the case if the first 2^k $\boldsymbol{x}$'s all map to the same coset, and this repeated for all 2^{n-k} cosets. Algorithms 1 and 2 comprehend a literal transcript of Figure 4 and compute the lower bound and upper bound respectively. Auxiliary variables $s^{\boldsymbol{p}}$ and $s^{\boldsymbol{x}}$ accumulate black and gray squares respectively. To maintain generality, we abstain from special case algorithms for linear codes, although it would result in a few simplifications.

Algorithm 1: BoundWorstCase

Input: List $\langle |\varphi_j|, q_j \rangle$
Output: Lower bound on $\widetilde{\mathbb{H}}_\infty(Y|P)$
$j, q, s^{\boldsymbol{p}} \leftarrow 0$
while $s^{\boldsymbol{p}} < 2^n$ **do**
  $j \leftarrow j + 1$
  $s_j^{\boldsymbol{p}} \leftarrow \min(|\varphi_j||\mathcal{M}|, 2^n - s^{\boldsymbol{p}})$
  $s^{\boldsymbol{p}} \leftarrow s^{\boldsymbol{p}} + s_j^{\boldsymbol{p}}$
  $q \leftarrow q + s_j^{\boldsymbol{p}} \cdot q_j$
$\widetilde{\mathbb{H}}_\infty(Y|P) \leftarrow -\log_2(q/|\mathcal{M}|)$

Algorithm 2: BoundBestCase

Input: List $\langle |\varphi_j|, q_j \rangle$
Output: Upper bound on $\widetilde{\mathbb{H}}_\infty(Y|P)$
$j, q, s^{\boldsymbol{p}}, s^{\boldsymbol{x}} \leftarrow 0$
while $s^{\boldsymbol{p}} < 2^n$ **do**
  $j \leftarrow j + 1$
  $s^{\boldsymbol{x}} \leftarrow s^{\boldsymbol{x}} + |\varphi_j|$
  $s_j^{\boldsymbol{p}} \leftarrow \lceil (s^{\boldsymbol{x}} - s^{\boldsymbol{p}})/|\mathcal{M}| \rceil |\mathcal{M}|$
  $s_j^{\boldsymbol{p}} \leftarrow \min(\max(s_j^{\boldsymbol{p}}, 0), 2^n - s^{\boldsymbol{p}})$
  $s^{\boldsymbol{p}} \leftarrow s^{\boldsymbol{p}} + s_j^{\boldsymbol{p}}$
  $q \leftarrow q + s_j^{\boldsymbol{p}} \cdot q_j$
$\widetilde{\mathbb{H}}_\infty(Y|P) \leftarrow -\log_2(q/|\mathcal{M}|)$

Algorithms 1 and 2 may now be applied to a variety of distributions. For a uniform distribution, the lower and upper bound both evaluate to $\widetilde{\mathbb{H}}_\infty(Y|P) = log_2(|\mathcal{M}|)$, regardless of other code specifics. Or simply k, for block codes in particular. The min-entropy loss is hence exactly $(n-k)$, given that $\mathbb{H}_\infty(X) = n$. Reyzin's proof [20] therefore implicates that the general-purpose $(n-k)$ bound cannot be tightened any further. Although results are fairly presentable already for the biased and correlated distributions, we further tighten these bounds first.

5.3 Tighter Bounds

Tighter bounds can be obtained by leveraging code properties more effectively. Algorithms 3 and 4 generalize Algorithms 1 and 2 respectively. In the former case, an additional input imposes an upper bound on the accumulated number of black squares, i.e., $\forall j, (s_1^{\boldsymbol{p}} + s_2^{\boldsymbol{p}} + \ldots + s_j^{\boldsymbol{p}}) \leq (u_1^{\boldsymbol{p}} + u_2^{\boldsymbol{p}} + \ldots + u_j^{\boldsymbol{p}})$. In the latter case, an additional input imposes a lower bound on the accumulated number of black squares, i.e., $\forall j, (s_1^{\boldsymbol{p}} + s_2^{\boldsymbol{p}} + \ldots + s_j^{\boldsymbol{p}}) \geq (l_1^{\boldsymbol{p}} + l_2^{\boldsymbol{p}} + \ldots + l_j^{\boldsymbol{p}})$. We now provide several examples.

Algorithm 3: BoundWorstCase2

Input: List $\langle |\varphi_j|, q_j, u_j^{\boldsymbol{p}} \rangle$
Output: Lower bound on $\widetilde{\mathbb{H}}_\infty(Y|P)$
$j, q, s^{\boldsymbol{p}}, u^{\boldsymbol{p}} \leftarrow 0$
while $s^{\boldsymbol{p}} < 2^n$ **do**
  $j \leftarrow j + 1$
  $u^{\boldsymbol{p}} \leftarrow u^{\boldsymbol{p}} + u_j^{\boldsymbol{p}}$
  $s_j^{\boldsymbol{p}} \leftarrow \min(|\varphi_j||\mathcal{M}|, u^{\boldsymbol{p}} - s^{\boldsymbol{p}})$
  $s_j^{\boldsymbol{p}} \leftarrow \min(s_j^{\boldsymbol{p}}, 2^n - s^{\boldsymbol{p}})$
  $s^{\boldsymbol{p}} \leftarrow s^{\boldsymbol{p}} + s_j^{\boldsymbol{p}}$
  $q \leftarrow q + s_j^{\boldsymbol{p}} \cdot q_j$
$\widetilde{\mathbb{H}}_\infty(Y|P) \leftarrow -\log_2(q/|\mathcal{M}|)$

Algorithm 4: BoundBestCase2

Input: List $\langle |\varphi_j|, q_j, l_j^{\boldsymbol{p}} \rangle$
Output: Upper bound on $\widetilde{\mathbb{H}}_\infty(Y|P)$
$j, q, s^{\boldsymbol{p}}, s^{\boldsymbol{x}}, l^{\boldsymbol{p}} \leftarrow 0$
while $s_{1:j}^{\boldsymbol{p}} < 2^n$ **do**
  $j \leftarrow j + 1$
  $s^{\boldsymbol{x}} \leftarrow s^{\boldsymbol{x}} + |\varphi_j|$
  $l^{\boldsymbol{p}} \leftarrow l^{\boldsymbol{p}} + l_j^{\boldsymbol{p}}$
  $s_j^{\boldsymbol{p}} \leftarrow \lceil (s^{\boldsymbol{x}} - s^{\boldsymbol{p}})/|\mathcal{M}| \rceil |\mathcal{M}|$
  $s_j^{\boldsymbol{p}} \leftarrow \max(s_j^{\boldsymbol{p}}, l^{\boldsymbol{p}} - s^{\boldsymbol{p}}, 0)$
  $s_j^{\boldsymbol{p}} \leftarrow \min(s_j^{\boldsymbol{p}}, 2^n - s^{\boldsymbol{p}})$
  $s^{\boldsymbol{p}} \leftarrow s^{\boldsymbol{p}} + s_j^{\boldsymbol{p}}$
  $q \leftarrow q + s_j^{\boldsymbol{p}} \cdot q_j$
$\widetilde{\mathbb{H}}_\infty(Y|P) \leftarrow -\log_2(q/|\mathcal{M}|)$

Worst-Case Bounds We improve the lower bound on $\widetilde{\mathbb{H}}_\infty(Y|P)$ for the correlated distribution. At least, for linear codes having the all-ones vector $\mathbf{1}$ of length n as a codeword. This includes Reed-Muller codes of any order [16]. This also includes many BCH, Hamming and repetition codes, on the condition that these are cyclic and having d odd, as easily proven hereafter. Consider an arbitrary codeword with Hamming weight d. XORing all 2^n circular shifts of this codeword results in the all-ones codeword, which ends the proof. As mentioned before, each set φ_j of the correlated distribution can be partitioned in pairs

$\{\boldsymbol{x}, \overline{\boldsymbol{x}}\}$, with $\overline{\boldsymbol{x}}$ the ones' complement of $\boldsymbol{x}$. Paired inputs belong to the same coset, i.e., maximum overlap in terms of helper data $\boldsymbol{p}$. Therefore, we impose the cumulative upper bound given by Equation (17).

$$u_j^{\boldsymbol{p}} = |\mathcal{M}|\frac{|\varphi_j|}{2} = 2^{k-1}|\varphi_j|. \tag{17}$$

For instance, consider linear/cyclic $[n, k = 1, d = n]$ repetition codes, i.e., having generator matrix $\boldsymbol{G} = \mathbf{1}$, with n odd. Algorithms BoundWorstCase2 and BoundBestCase then converge to the exact result $\widetilde{\mathbb{H}}_\infty(Y|P) = 1$, not depending on parameter c. This is the best-case scenario, given the universal bound $\widetilde{\mathbb{H}}_\infty(Y|P) \leq k$. Figure 6 illustrates the former with squares for $n = 5$. The result also holds if the repetition code is neither linear/cyclic nor odd. As long as $\boldsymbol{w}_1 \oplus \boldsymbol{w}_2 = \mathbf{1}$, the elements of each φ_j can be paired into cosets. Although the term coset is usually preserved for linear codes, translations of a non-linear repetition code are either disjunct or coincide and still partition the space $\{0,1\}^{1\times n}$.

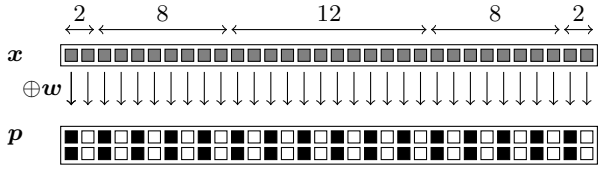


Fig. 6. The exact residual min-entropy $\widetilde{\mathbb{H}}_\infty(Y|P)$ for the correlated distribution and an $[n = 5, k = 1, d = 5]$ repetition code.

Best-Case Bounds We improve the upper bound on $\widetilde{\mathbb{H}}_\infty(Y|P)$ for both the biased and correlated distribution. In particular, we take minimum distance d into account. The main insight is that two slightly differing inputs $\boldsymbol{x}_u \neq \boldsymbol{x}_v$ do not overlap in terms of helper data $\boldsymbol{p}$. More precisely, if $\mathsf{HD}(\boldsymbol{x}_u, \boldsymbol{x}_v) \in [1, d-1]$, then $\{\boldsymbol{x}_u \oplus \boldsymbol{w} \mid \boldsymbol{w} \in \mathcal{W}\} \cap \{\boldsymbol{x}_v \oplus \boldsymbol{w} \mid \boldsymbol{w} \in \mathcal{W}\} = \varnothing$. For the biased distribution, the following holds: $\mathsf{HD}(\boldsymbol{x}_u, \boldsymbol{x}_v) \in [1, d-1]$ if $\boldsymbol{x}_u \neq \boldsymbol{x}_v$ and $\boldsymbol{x}_u, \boldsymbol{x}_v \in (\varphi_1 \cup \varphi_2 \cup \ldots \cup \varphi_{t+1})$. Or stated otherwise, the elements of the first $t+1$ sets all result in unique $\boldsymbol{p}$'s. Therefore, we can impose the constraint given by Equation (18). Figure 7 depicts the squares.

$$l_j^{\boldsymbol{p}} = \begin{cases} |\varphi_j||\mathcal{M}|, & \text{if } j \in [1, t+1] \\ 0, & \text{otherwise} \end{cases}. \tag{18}$$

There is a remarkable observation for perfect codes in particular. As clear from the Hamming bound in Equation (4), all $\boldsymbol{p}$'s are covered by the first $t+1$ sets exclusively. BoundWorstCase and BoundBestCase2 hence produce the same output. I.e., an exact evaluation of the residual min-entropy, as further simplified by Equation (19). With F_B, we denote the cumulative distribution function of a

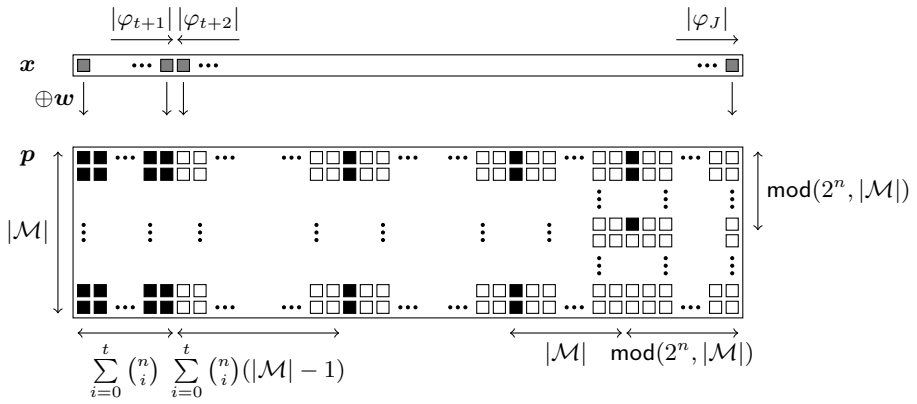


Fig. 7. A tightened upper bound on $\widetilde{\mathbb{H}}_\infty(Y|P)$ for the biased distribution, hereby making use of Equation (18).

binomial distribution. This considerable extends a prior result of Delvaux et al. in [8]. The same formula was derived for $[n, k = 1, d = n]$ repetition codes, with n odd. Note that such repetition codes are perfect. As a side note, their result was proven for the methods of Bennett et al. and Dodis et al. separately. Our equivalencies in Section 4 indicate that latter by itself would have been sufficient.

$$\widetilde{\mathbb{H}}_\infty(Y|P) = -\log_2\left(\sum_{j=1}^{t+1} |\varphi_j| \cdot q_j\right) = -\log_2(\mathsf{F}_B(t; n, \min(b, 1-b))). \tag{19}$$

For codes which do not happen to be perfect, there is still margin for improvement. We inject some promising thoughts but abstain from numerical results later-on. Consider a linear code of which the Hamming weight distribution of the coset leaders $\boldsymbol{\epsilon}$ is well-understood. Let $|\mathcal{E}_h|$ denote the number of cosets such that $h = \mathsf{HW}(\boldsymbol{\epsilon})$. Clearly, $|\mathcal{E}_h| = \binom{n}{h}$ for $h \in [0, t]$. Our interest concerns $|\mathcal{E}_h|$ for $h > t$, all of which are exactly known in the ideal case, as in [6] for certain BCH codes. The largest h for which $|\mathcal{E}_h| > 0$ is also referred to as the *covering radius* h_{cr} of the code. For a bias $b < \frac{1}{2}$, Equation (20) comprehends the exact residual min-entropy. The latter expression extends to $b > \frac{1}{2}$ in case the all-ones vector $\mathbf{1}$ is a codeword. This includes Reed-Muller codes as well as cyclic codes with d odd, as has been argued earlier-on. If only bounds on $|\mathcal{E}_h|$ and/or h_{cr} are known, one might still be able to further tighten the bounds on $\widetilde{\mathbb{H}}_\infty(Y|P)$ correspondingly.

$$\widetilde{\mathbb{H}}_\infty(Y|P) = -\log_2\left(\frac{1}{|\mathcal{M}|}\sum_{h=0}^{h_{\mathsf{cr}}} |\mathcal{E}_h| \cdot |\mathcal{M}| \cdot q_{h+1}\right) = -\log_2\left(\sum_{h=0}^{h_{\mathsf{cr}}} |\mathcal{E}_h| \cdot q_{h+1}\right). \tag{20}$$

For instance, consider $[n, k = 1, d = n]$ repetition codes with n even. These form the non-perfect and therefore less popular counterpart of n odd. Inputs $\boldsymbol{x}$ belonging to φ_j and φ_{n+2-j} are still paired in order to form the cosets. Unlike n odd, there is a central set φ_{t+2} which contains both members of each pair. Therefore, $h_{\mathsf{cr}} = t + 1$ and $|\mathcal{E}_{t+1}| = |\varphi_{t+2}|/2$. As argued before, the operational principles of cosets extend to non-linear repetition codes. Figure 8 depicts the squares for $n = 4$. Equation (21) evaluates the residual min-entropy.

$$\widetilde{\mathbb{H}}_\infty(Y|P) = -\log_2\left(\mathsf{F}_B(t; n, \min(b, 1-b)) + \frac{1}{2}\binom{n}{\frac{n}{2}}(b(1-b))^{\frac{n}{2}}\right). \tag{21}$$

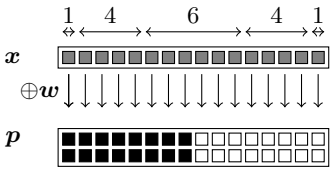


Fig. 8. The exact residual min-entropy $\widetilde{\mathbb{H}}_\infty(Y|P)$ for the biased distribution and an $[n = 4, k = 1, d = 4]$ repetition code.

Also for the correlated distribution, distance d might be incorporated to tighten the upper bound on $\widetilde{\mathbb{H}}_\infty(Y|P)$. First of all, we assign $|\mathcal{M}|$ unique $\boldsymbol{p}$'s to one out of two elements in φ_1. For ease of understanding, assume $\boldsymbol{x} = \mathbf{0}$, comprehending the first case in Equation (22). For each set φ_j, with $j \in [2, n]$, we then count the number of inputs $\boldsymbol{x} \in \varphi_j$ such that $h = \mathsf{HW}(\boldsymbol{x}) \leq t$. The latter constraint guarantees all assigned $\boldsymbol{p}$'s to be unique. We distinguish between two forms, $\boldsymbol{x} = (\mathbf{0}\|\mathbf{1}\|\mathbf{0}\|\ldots)$ and $\boldsymbol{x} = (\mathbf{1}\|\mathbf{0}\|\mathbf{1}\|\ldots)$, resulting in two main terms. For each form, we apply *stars and bars* combinatorics twice. In particular, we assign h indistinguishable stars, i.e., ones, to distinguishable bins and independently also for $n - h$ zeros. Note that $l_j^{\boldsymbol{p}} = 0$ for $j > 2t + 1$. To ensure formula correctness, one may verify numerically that $l_1^{\boldsymbol{p}} + l_2^{\boldsymbol{p}} + \ldots + l_{2t+1}^{\boldsymbol{p}}$ equals the left hand side of the Hamming bound in Equation (4).

$$l_j^{\boldsymbol{p}} = \begin{cases} |\mathcal{M}|, & \text{if } j = 1 \\ |\mathcal{M}|\Big(\sum_{h=\lfloor j/2 \rfloor}^{t} \binom{h-1}{\lfloor j/2 \rfloor - 1}\binom{n-h-1}{\lceil j/2 \rceil - 1} & \\ \quad + \sum_{h=\lceil j/2 \rceil}^{t} \binom{h-1}{\lceil j/2 \rceil - 1}\binom{n-h-1}{\lfloor j/2 \rfloor - 1}\Big), & \text{otherwise.} \end{cases} \tag{22}$$

5.4 Numerical Results

Figure 9 presents numerical results for various BCH codes. We focus on small codes, as these allow for an exact exhaustive evaluation of the residual min-entropy using Equation (13) and/or (14). As such, the tightness of various

bounds can be assessed adequately. Figure 9(d) nevertheless demonstrates that our algorithms support large codes equally well, in compliance with a practical key generator. Note that only half of the bias interval $b \in [0, 1]$ is depicted. The reason is that all curves mirror around the vertical axis of symmetry $b = \frac{1}{2}$. The same holds for the correlated distribution with parameter c.

Especially the lower bounds perform well, which benefits a conservative system provider. The best lower bounds in Figures 9(a), (b) and (c) visually coincide with the exact result. The gap with the $(n - k)$ bound is the most compelling around $b, c \approx 0.7$, where the corresponding curves hit the horizontal axis $\widetilde{\mathbb{H}}_\infty(Y|P) = 0$. Also our upper bounds are considerably tighter than their more general alternatives in Equation (11). Nevertheless, the latter bounds remain open for further improvement, with the exception of Figure 9(b). An $[n = 7, k = 4, d = 3]$ code is perfect and lower and upper bounds then converge to the exact result for a biased distribution.

5.5 Exhaustive Tightening

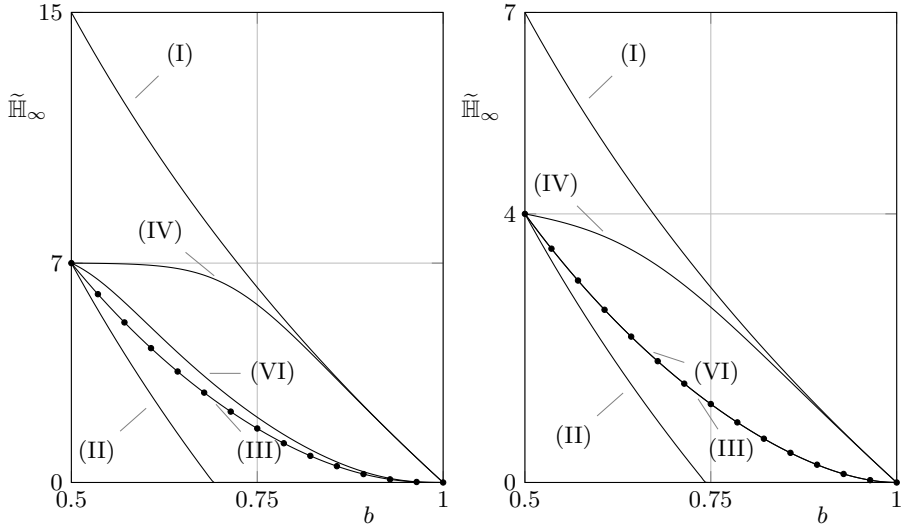
An exhaustive evaluation of Equations (13) and (14) was deemed infeasible in practice due to large codes. Nevertheless, to the extent possible, number crunching may further tighten the outcome of Algorithms 3 and 4. In particular, we would target the initial sets, i.e., φ_1 up to a certain φ_g, as these contribute relatively the most to $\widetilde{\mathbb{H}}_\infty(X|P)$. The assignment of black and white squares is then exact. Starting from set φ_{g+1}, the usual computations take over. We stress that linear codes are by far the most suitable for this hybrid technique.

6 Conclusion

Secure sketches are the main workhorse of modern PUF-based key generators. The min-entropy loss of most sketches is upper-bounded by $(n - k)$ bits and designers typically instantiate system parameters accordingly. However, the latter bound tends to be overly pessimistic, resulting in an unfortunate implementation overhead. We showcased the proportions for a prominent category of PUFs, with bias and spatial correlations acting as the main non-uniformities. New considerably tighter bounds were derived, valid for a variety of popular but algebraically complex codes. These bounds are unified in the sense of being applicable to seven secure sketch constructions. Deriving tighter alternatives for the $(n - k)$ bound counts as unexplored territory and we not claim to have reached the end of it. For instance, new techniques may have to be developed in order to tackle more advanced *second-order* distributions.

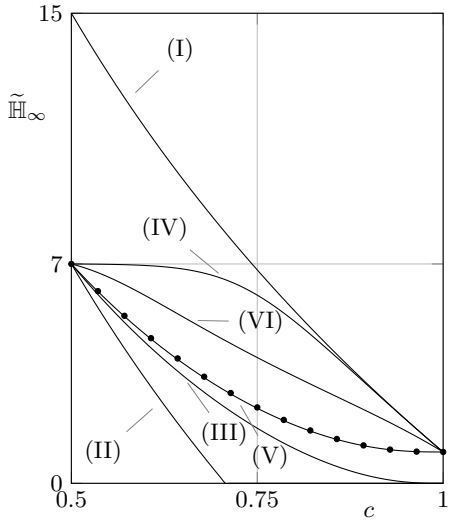
Acknowledgment

The authors greatly appreciate the support received. The European Union's Horizon 2020 research and innovation programme under grant number 644052

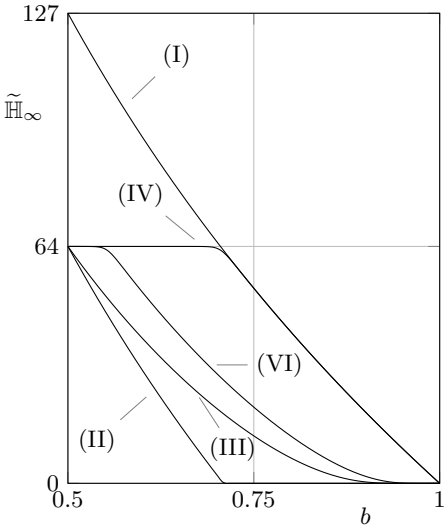


(a) Bias; $[n = 15, k = 7, d = 5]$.

(b) Bias; $[n = 7, k = 4, d = 3]$.



(c) Correlation; $[n = 15, k = 7, d = 5]$.



(d) Bias; $[n = 127, k = 64, d = 21]$.

Fig. 9. Secure sketch min-entropy loss for various BCH codes. Dots correspond to an exact exhaustive evaluation of Equation (13)/(14). The legend of the curves is as follows. (I) The ingoing min-entropy $\mathbb{H}_\infty(X) = -\log_2(q_1)$. (II) The lower bound $\widetilde{\mathbb{H}}_\infty(X|P) = \max(\mathbb{H}_\infty(X) - (n-k), 0)$. (III) The lower bound on $\widetilde{\mathbb{H}}_\infty(X|P)$ according to BoundWorstCase. (IV) The upper bound on $\widetilde{\mathbb{H}}_\infty(X|P)$ according to BoundBestCase. (V) The lower bound on $\widetilde{\mathbb{H}}_\infty(X|P)$ according to BoundWorstCase2. (VI) The upper bound on $\widetilde{\mathbb{H}}_\infty(X|P)$ according to BoundBestCase2.

(HECTOR). The Research Council of KU Leuven, GOA TENSE (GOA/11/007), the Flemish Government through FWO G.0550.12N and the Hercules Foundation AKUL/11/19. The national major development program for fundamental research of China (973 Plan) under grant number 2013CB338004. Jeroen Delvaux is funded by IWT-Flanders grant number SBO 121552. Matthias Hiller is funded by the German Federal Ministry of Education and Research (BMBF) in the project SIBASE through grant number 01IS13020A.

References

1. R. Ahlswede and I. Csiszár. Common Randomness in Information Theory and Cryptography - Part I: Secret Sharing. *IEEE Transactions on Information Theory*, 39(4):1121–1132, 1993.
2. B. Barak, Y. Dodis, H. Krawczyk, O. Pereira, K. Pietrzak, F. Standaert, and Y. Yu. Leftover Hash Lemma, Revisited. In *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference*, pages 1–20, 2011.
3. C. H. Bennett, G. Brassard, C. Crépeau, and M. Skubiszewska. Practical Quantum Oblivious Transfer. In *Advances in Cryptology - CRYPTO 1991, 11th Annual Cryptology Conference*, pages 351–366, 1991.
4. C. Bösch, J. Guajardo, A. Sadeghi, J. Shokrollahi, and P. Tuyls. Efficient Helper Data Key Extractor on FPGAs. In *Cryptographic Hardware and Embedded Systems - CHES 2008, 10th International Workshop*, pages 181–197, 2008.
5. L. Carter and M. N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18(2):143–154, 1979.
6. P. Charpin, T. Helleseth, and V. A. Zinoviev. The Coset Distribution of Triple-Error-Correcting Binary Primitive BCH Codes. *IEEE Transactions on Information Theory*, 52(4):1727–1732, 2006.
7. G. I. Davida, Y. Frankel, and B. J. Matt. On Enabling Secure Applications Through Off-Line Biometric Identification. In *Security and Privacy - 1998 IEEE Symposium on Security and Privacy*, pages 148–157, 1998.
8. J. Delvaux, D. Gu, D. Schellekens, and I. Verbauwhede. Helper Data Algorithms for PUF-Based Key Generation: Overview and Analysis. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 2015.
9. Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. *SIAM Journal on Computing*, 38(1):97–139, 2008.
10. W. Feller. *An Introduction to Probability Theory and Its Applications, Vol. 1, 3rd Edition.* 1968.
11. J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A Pseudorandom Generator from any One-way Function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.
12. M. Hiller, M. Yu, and M. Pehl. Systematic Low Leakage Coding for Physical Unclonable Functions. In *ASIA CCS 2015, 10th ACM Symposium on Information, Computer and Communications Security*, pages 155–166, 2015.
13. 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*, 58(9):1198–1210, 2009.
14. A. Juels and M. Wattenberg. A Fuzzy Commitment Scheme. In *CCS 1999, 6th ACM Conference on Computer and Communications Security*, pages 28–36, 1999.

15. H. Kang, Y. Hori, T. Katashita, M. Hagiwara, and K. Iwamura. Cryptographic Key Generation from PUF Data Using Efficient Fuzzy Extractors. In *ICACT 2014, 16th International Conference on Advanced Communication Technology*, 2014.
16. F. J. MacWiliams and N. J. A. Sloane. *The theory of error correcting codes*. 1977.
17. R. Maes, P. Tuyls, and I. Verbauwhede. A Soft Decision Helper Data Algorithm for SRAM PUFs. In *ISIT 2009, IEEE International Symposium on Information Theory*, pages 2101–2105, 2009.
18. R. Maes, V. van der Leest, E. van der Sluis, and F. Willems. Secure Key Generation from Biased PUFs. Cryptology ePrint Archive, Report 2015/583, 2015. http://eprint.iacr.org/ (CHES 2015).
19. R. Maes, A. Van Herrewege, and I. Verbauwhede. PUFKY: A Fully Functional PUF-Based Cryptographic Key Generator. In *Cryptographic Hardware and Embedded Systems - CHES 2012 - 14th International Workshop*, pages 302–319, 2012.
20. L. Reyzin. Entropy Loss is Maximal for Uniform Inputs. Technical Report BUCS-TR-2007-011, Department of Computer Science, Boston University, September 2007.
21. Y. Sutcu, Q. Li, and N. D. Memon. Protecting Biometric Templates With Sketch: Theory and Practice. *IEEE Transactions on Information Forensics and Security*, 2(3-2):503–512, 2007.
22. P. Tuyls, A. H. M. Akkermans, T. A. M. Kevenaar, G.-J. Schrijen, A. M. Bazen, and R. N. J. Veldhuis. Practical Biometric Authentication with Template Protection. In *AVBPA 2005, Int. Conference on Audio- and Video-Based Biometric Person Authentication*, pages 436–446, 2005.
23. P. Tuyls, G.-J. Schrijen, B. Škorić, J. van Geloven, N. Verhaegh, and R. Wolters. Read-Proof Hardware from Protective Coatings. In *CHES 2006, Int. Workshop on Cryptographic Hardware and Embedded Systems*, pages 369–383, 2006.
24. Y. Wang, S. Rane, S. C. Draper, and P. Ishwar. A Theoretical Analysis of Authentication, Privacy, and Reusability Across Secure Biometric Systems. *IEEE Transactions on Information Forensics and Security*, 7(6):1825–1840, 2012.
25. H. Yu, P. H. W. Leong, H. Hinkelmann, L. Möller, M. Glesner, and P. Zipf. Towards a Unique FPGA-Based Identification Circuit Using Process Variations. In *FPL 2009, Int. Conference on Field Programmable Logic and Applications*, pages 397–402, 2009.
26. M. Yu. Turn FPGAs Into "Key" Players In The Cryptographics Field, Jul 2009. Electronic Design Magazine, http://electronicdesign.com/fpgas/turn-fpgas-key-players-cryptographics-field.
27. M. Yu and S. Devadas. Secure and Robust Error Correction for Physical Unclonable Functions. *IEEE Design & Test of Computers*, 27(1):48–65, 2010.

A Fuzzy Extractors

The *statistical distance* between two probability distributions, X_1 and X_2, is defined as in Equation (23).

$$\mathsf{SD}(X,Y) = \frac{1}{2}\Big|\sum_{x\in(\mathcal{X}_1\cup\mathcal{X}_2)}(\mathbb{P}(X_1=x)-\mathbb{P}(X_2=x))\Big|. \tag{23}$$

An *average-case fuzzy extractor* [9] is a pair of efficient and possibly randomized procedures: the generation procedure $\boldsymbol{p} \leftarrow \mathsf{FEGen}(x)$, with helper data

$\boldsymbol{p} \in \mathcal{P}$, and the recovery procedure $\boldsymbol{k} \leftarrow \mathsf{FERep}(\widetilde{x}, \boldsymbol{p})$, with key $\boldsymbol{k} \in \mathcal{K}$. There are two defining properties, as listed below.

- *Correctness.* If $\mathsf{dist}(x, \widetilde{x}) \leq t$, correctness of reconstruction is guaranteed, i.e., $\mathsf{FERep}(x, \boldsymbol{p}) = \mathsf{FERep}(\widetilde{x}, \boldsymbol{p})$. If $\mathsf{dist}(x, \widetilde{x}) > t$, there is no guarantee whatsoever.
- *Security.* For a certain lower-bound on the ingoing min-entropy, i.e., $\widetilde{\mathbb{H}}_\infty(X|I) \geq h_{in}$, the string $\boldsymbol{k}$ is guaranteed to be nearly uniform, even if the helper data is observed. Formally, $\mathsf{SD}((K, P, I), (U, P, I)) \leq \epsilon$, with U uniformly distributed. It is advisable to choose ϵ negligibly small, e.g., 2^{-128}.

There is a proven standard method to craft a fuzzy extractor from a secure sketch. In particular, a *randomness extractor* could derive a key from the secure sketch output, i.e., $\boldsymbol{k} \leftarrow \mathsf{Ext}(y)$. *Universal hash functions* [5] are good randomness extractors, according to the *(generalized) leftover hash lemma* [11, 2]. Unfortunately, their min-entropy loss is still quite substantial. In practice, key generators therefore often rely on a cryptographic hash function which is assumed to behave as a *random oracle*. The latter idealized heuristic results in zero min-entropy loss.

B The Sketch of Davida et al. and its Performance

The method of Davida et al. [7] is represented by Figure 10. It requires a linear code $\mathcal{C}$ with the generator matrix in standard form, i.e., $\boldsymbol{G} = (\boldsymbol{I}_k \| \boldsymbol{A})$. The method can be understood as an unmasked version of Kang et al. in Figure 1(f). Equation (3) implies that the min-entropy loss $\Delta\mathbb{H}_\infty$ is bounded by $(n - k)$ bits, as before. This is relatively large though, as sketch input $\boldsymbol{x}$ comprehends k rather than n bits, in contrast to all previous methods. On the bright side, t is relatively larger as well, as only k bits are prone to error. Juels et al. [14] labelled their sketch nevertheless as an absolute improvement, implying that the latter does not fully compensate for the former. We are the first to support this claim with an extensive quantitative analysis.

$\boldsymbol{p} \leftarrow \mathsf{SSGen}(\boldsymbol{x})$	$\widehat{\boldsymbol{y}} \leftarrow \mathsf{SSRep}(\widetilde{\boldsymbol{x}}, \boldsymbol{p})$	
$\boldsymbol{p} \leftarrow \boldsymbol{x} \cdot \boldsymbol{A}$	$\widehat{\boldsymbol{y}} = \widehat{\boldsymbol{x}} \leftarrow \mathsf{Decode}(\widetilde{\boldsymbol{x}} \| \boldsymbol{p})$	Systematic method of Davida et al. [7].

Fig. 10. The secure sketch construction of Davida et al., having a k-bit input $\boldsymbol{x}$. Correctness is guaranteed, given a noisy version $\widetilde{\boldsymbol{x}}$ with $\mathsf{HD}(\boldsymbol{x}, \widetilde{\boldsymbol{x}}) \leq t$.

Given the equivalencies in Section 4, the performance upgrade actually applies to all seven sketches. We limit the scope of our comparison to a uniformly distributed X with $\mathcal{I}$ empty. As proven in Section 5.2, the equivalent sketches

then have a residual min-entropy $\widetilde{\mathbb{H}}_\infty(X|P) = k$. Similarly, for the sketch of Davida et al., we obtain Equation (24) and subsequently also Equation (25). When considering $\boldsymbol{A}$ as a linear map from $\mathcal{X}$ to $\mathcal{P}$, it follows that $|\mathcal{P}| = 2^{\mathrm{rank}(\boldsymbol{A})}$.

$$\mathbb{P}((P = \boldsymbol{p})|(X = \boldsymbol{x})) = \begin{cases} 1, & \text{if } \boldsymbol{p} = \boldsymbol{x} \cdot \boldsymbol{A} \\ 0, & \text{otherwise} \end{cases}. \tag{24}$$

$$\widetilde{\mathbb{H}}_\infty(X|P) = -\log_2\left(|\mathcal{P}|\frac{1}{2^k}\right) = -\log_2\left(2^{\mathrm{rank}(\boldsymbol{A})}\frac{1}{2^k}\right) = k - \mathrm{rank}(\boldsymbol{A}). \tag{25}$$

The rank of a matrix is nonnegative and upper bounded by its dimensions, i.e., $0 \leq \mathrm{rank}(\boldsymbol{A}) \leq \min(k, n-k)$. In practice though, $\boldsymbol{A}$ tends to have full rank, leading to Equation (26). Rank deficiencies would reduce the potential for a high t. In the extreme case of a zero matrix, i.e., $\mathrm{rank}(\boldsymbol{A}) = 0$, there is no min-entropy loss, but we would end up with $t = 0$. For cyclic codes, $\boldsymbol{A}$ is guaranteed to have full rank, as proven next. Applying a circular shift to the columns of $G = (\boldsymbol{I}_k \| \boldsymbol{A})$ results in an equivalent generator matrix $G' = (\boldsymbol{A} \| \boldsymbol{I}_k)$. The latter matrix could equally well have been obtained via rank-preserving elementary row operations. The number of non-zero rows in the submatrix consisting of the rightmost $(n-k)$ columns of G' equals $\min(k, n-k)$, i.e., the rank of $\boldsymbol{A}$, which ends the proof. Note that it is of crucial importance to use codes with more message bits than redundancy bits, i.e., $k > (n-k)$. For, e.g., repetition codes, there would be zero min-entropy left.

$$\widetilde{\mathbb{H}}_\infty(X|P) = \max(0, 2k - n). \tag{26}$$

As an ultimate performance comparison, we instantiate key generators under the following specification. We assume a uniform bit error rate, i.e., $P_E = 5\%$. Furthermore, we aim for $\widetilde{\mathbb{H}}_\infty(X|P) = 128$. Finally, we impose a failure rate $P_F = 10^{-6}$ on the key reconstruction. For a given number of parallel code instances z, in accordance with Section 4.6, we instantiate with the smallest parameters $[n_1, k_1, t_1]$ which meet the specifications. For Davida et al., $P_F = 1 - (\mathsf{F}_B(t; k, P_E))^z$; for all other sketches, $P_F = 1 - (\mathsf{F}_B(t; n, P_E))^z$. We use optimal codes. Although not necessarily suitable for implementation, these capture perfectly how code parameters scale when codes get bigger. We obtain the numbers in Table 2. Each row is part of a Pareto frontier, graphically represented in Figure 11. The sketch of Davida et al. is clearly inefficient in terms of code complexity and the number of ingoing bits, i.e., $\mathbb{H}_\infty(X)$.

C Secure Sketch Equivalencies for Shannon Entropy

The sketch equivalencies proven for min-entropy in Section 4 trivially extend to Shannon entropy. Therefore, we limit ourselves to replacing formulas. Equations (27) and (28) define Shannon entropy and conditional Shannon entropy

		Equivalent sketches			Davida et al.		
z	$\widetilde{\mathbb{H}}_\infty(X\|P)$	$[n_1, k_1, t_1]$	$\mathbb{H}_\infty(X)$	P_F	$[n_1, k_1, t_1]$	$\mathbb{H}_\infty(X)$	P_F
1	128	$[269, 128, 33]$	269	$\approx 8.15 \cdot 10^{-7}$	$[502, 315, 37]$	315	$\approx 6.26 \cdot 10^{-7}$
2	128	$[158, 64, 24]$	316	$\approx 7.10 \cdot 10^{-7}$	$[336, 200, 28]$	400	$\approx 5.90 \cdot 10^{-7}$
3	129	$[121, 43, 21]$	363	$\approx 4.43 \cdot 10^{-7}$	$[271, 157, 24]$	471	$\approx 9.41 \cdot 10^{-7}$
4	128	$[99, 32, 19]$	396	$\approx 3.53 \cdot 10^{-7}$	$[246, 139, 23]$	556	$\approx 4.55 \cdot 10^{-7}$

Table 2. Performance of all equivalent sketches versus Davida et al.

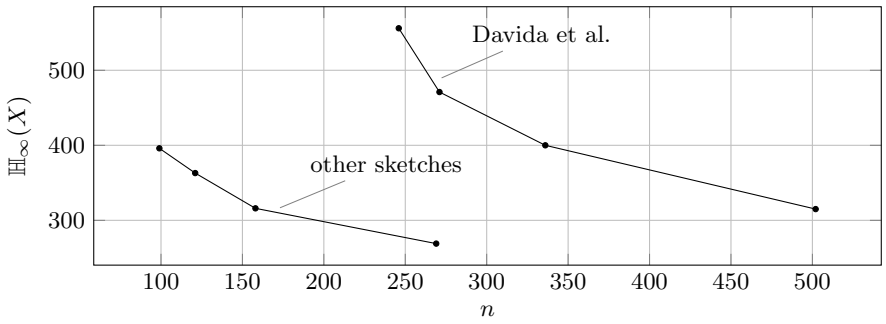


Fig. 11. Pareto frontiers. We choose n as a representative for the code complexity.

respectively, similar to Equations (1) and (2) before. Undefined terms $0 \cdot \log_2(0)$ are evaluated as 0. The same holds for terms with $\mathbb{P}(I = i) = 0$.

$$\mathbb{H}(X) = -\mathbb{E}_{x \leftarrow X}\big[\log_2(\mathbb{P}(X = x))\big]. \tag{27}$$

$$\widetilde{\mathbb{H}}(X|I) = -\mathbb{E}_{i \leftarrow I}\Big[\sum_{x \in \mathcal{X}} \mathbb{P}((X = x)|(I = i)) \log_2(\mathbb{P}((X = x)|(I = i)))\Big]. \tag{28}$$

We argue that all seven sketches have the same residual Shannon entropy, as expressed by Equation (29). At least, given that the ingoing distribution (X, I) and underlying code $\mathcal{C}$ are the same. Equations (6), (7), (8) and (10) still apply. Equation (30) offers a replacement for Equation (9).

$$\begin{aligned}\widetilde{\mathbb{H}}(Y|(P, I)) = -\mathbb{E}_{(\boldsymbol{p}, i) \leftarrow (P, I)}\Big[\sum_{\boldsymbol{y} \in \mathcal{Y}} \mathbb{P}((Y = \boldsymbol{y})|((P = \boldsymbol{p}) \cap (I = i))) \\ \log_2(\mathbb{P}((Y = \boldsymbol{y})|((P = \boldsymbol{p}) \cap (I = i))))\Big].\end{aligned} \tag{29}$$

$$\begin{aligned}\widetilde{\mathbb{H}}(X|(P, I)) = -\mathbb{E}_{(\boldsymbol{\epsilon}, i) \leftarrow (E, I)}\Big[\sum_{\boldsymbol{w} \in \mathcal{C}} \mathbb{P}((X = \boldsymbol{\epsilon} \oplus \boldsymbol{w})|((E = \boldsymbol{\epsilon}) \cap (I = i))) \\ \log_2(\mathbb{P}((X = \boldsymbol{\epsilon} \oplus \boldsymbol{w})|((E = \boldsymbol{\epsilon}) \cap (I = i))))\Big].\end{aligned} \tag{30}$$