Efficient Fuzzy Extraction of PUF-Induced Secrets: Theory and Applications

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. The device-unique response of a *physically unclonable function* (PUF) can serve as the root of trust in an embedded cryptographic system. *Fuzzy extractors* transform this noisy non-uniformly distributed secret into a stable high-entropy key. The overall efficiency thereof, typically depending on error-correction with a binary $[n, k, d]$ block code, is determined by the universal and well-known $(n - k)$ bound on the min-entropy loss. We derive new considerably tighter bounds for PUF-induced distributions that suffer from, e.g., bias or spatial correlations. The bounds are easy-to-evaluate and apply to large non-trivial codes, e.g., BCH, Hamming and Reed-Muller codes. Apart from an inherent reduction in implementation footprint, the newly developed theory also facilitates the analysis of state-of-the-art error-correction methods for PUFs. As such, we debunk the reusability claim of the reverse fuzzy extractor. Moreover, we provide proper quantitative motivation for debiasing schemes, as this was missing in the original proposals.

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

1 Introduction

Cryptography relies on reproducible uniformly distributed secret keys. Obtaining affordable physically secure key-storage in embedded non-volatile memory is hard though. Harvesting entropy from *physically unclonable functions* (PUFs) comprehends an alternative that lowers the power-off state vulnerability. Unfortunately, PUF responses are corrupted by noise and non-uniformities are bound

to occur. A *fuzzy extractor* [14] provides an *information-theoretically secure* mechanism to convert PUF responses into high-quality keys. The essential building block for handling noisiness is the *secure sketch*, providing error-correction with a binary $[n, k, d]$ block code. Associated public helper data reveals information about the PUF response though; the designer should hence quantify how much min-entropy remains. So far, the conservative $(n-k)$ upper bound on the min-entropy loss has been applied. Unfortunately, the residual min-entropy is underestimated and more PUF response bits than necessary have to be used. Expensive die area is hence blocked by PUF circuits that are not strictly required to obtain the desired security level, i.e., symmetric key length.

1.1 Contribution

The novelty of our work is twofold:

- First, we derive new bounds on the secure sketch 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. The discrepancy is showcased for two predominant PUF imperfections, i.e., biased and spatially correlated response bits. It is important to note that a variety of commonly used code classes is covered, regardless of their algebraic complexity. Furthermore, a large variety of distributions could be supported. Therefore, our scope reaches considerably further than related work in [11, 27], 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.
- Second, we apply the newly developed theory to state-of-the-art error-correction methods for PUFs. As such, we reveal a fundamental flaw in the reverse fuzzy extractor, proposed by Van Herrewege et al. [33] at Financial Crypto 2012. The latter lightweight primitive is gaining momentum and has also been adopted in the CHES 2015 protocol of Aysu et al. [2]. We debunk the main security claim that repeated helper data exposure does not result in additional min-entropy loss. Furthermore, we contribute to the motivation of debiasing schemes such as the *index-based syndrome* (IBS) proposal of Yu et al. [37], and the CHES 2015 proposal of Maes et al. [27]. The latter proposals assume that a stand-alone sketch cannot handle biased distributions. We eliminate the need for an educated guess that originates from either the extrapolation of repetition code insights and/or the application of the overly conservative $(n-k)$ bound.

1.2 Organization

The remainder of this manuscript is organized as follows. Section 2 introduces notation and preliminaries. Section 3 derives new tight bounds on the secure sketch min-entropy loss. Second 4 elaborates applications of the newly developed theory. Section 5 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 $\mathbb{E}_{x \leftarrow X}[g(X)]$. The probability density function and cumulative distribution function of a standard normal distribution $N(0, 1)$ are denoted as $\mathsf{f}_{\mathsf{norm}}(\cdot)$ and $\mathsf{F}_{\mathsf{norm}}(\cdot)$ respectively. For a binomial distribution with n trials and success probability p, we use $\mathsf{f}_{\mathsf{bino}}(\cdot; n, p)$ and $\mathsf{F}_{\mathsf{bino}}(\cdot; n, p)$ respectively.

2.2 Min-Entropy Definitions

The *min-entropy* of a random variable X is as defined in (1). Consider now a pair of possibly correlated random variables: X and P. The *conditional min-entropy* [14] of X given P is as defined in (2). Terms with $\mathbb{P}(P = p) = 0$ are evaluated as 0. Both definitions quantify the probability that an attacker guesses $x \leftarrow X$ first time right, on a logarithmic scale. We emphasize that min-entropy is a more conservative notion than Shannon entropy and therefore often preferred within cryptology.

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

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

2.3 Physically Unclonable Functions

A prominent category of PUFs, suitable for key generation in particular, consists of an array of identically designed cells. Each cell produces a single bit, or occasionally a few bits. This includes memory-based designs, such as the SRAM PUF [19], as well as the coating PUF [31] and a subset of the large number of ring oscillator-based designs, e.g., [35]. 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.

We describe a parameterized probability distribution for the error rate of individual PUF response bits $\widetilde{\boldsymbol{x}}(i)$, with $i \in [1, n]$. Experimental validation on various PUF circuits, e.g., in [25, 13], labelled the model as accurate. Two hidden

random variables are incorporated: the normalized manufacturing variability $V_i \sim N(0,1)$, drawn only once for each response bit, and additive noise $N_{ij} \sim N(0,\sigma_N)$, drawn for each evaluation j of a given response bit. A response bit $\widetilde{\boldsymbol{x}}(i)$ evaluates to 1 if $(v_i+n_{ij}) > t$ and 0 vice versa, with threshold t a fixed parameter. Bias corresponds to a nonzero t. Spatial correlations can be incorporated via a multivariate normal distribution $\begin{pmatrix} V_1 & \dots & V_n \end{pmatrix} \sim N(\mathbf{0}, \boldsymbol{\Sigma})$, with $\boldsymbol{\Sigma}$ the symmetric $n \times n$ covariance matrix.

For ease of analysis, we consider the response bits $\boldsymbol{x}(i)$ obtained by thresholding $v_i > t$ as a reference. In practice, these nominal values can be approximated via a majority vote among noisy replicas $\widetilde{\boldsymbol{x}}(i)$, possibly accelerated via circuit techniques [6, 37]. Bias parameter b, defined as the probability $\mathbb{P}(\boldsymbol{x}(i) = 1)$, then equals $\mathsf{F}_{\mathsf{norm}}(-t)$. The error rate p_E of a response bit $\widetilde{\boldsymbol{x}}(i)$ with respect to its reference, i.e., the probability $\mathbb{P}(\boldsymbol{x}(i) \neq \widetilde{\boldsymbol{x}}(i))$, then equals $\mathsf{F}_{\mathsf{norm}}(-|v_i - t|/\sigma_N)$.

2.4 Secure Sketch and Fuzzy Extractor Definitions

Secure sketches operate on a metric space $\mathcal{X}$ with distance function dist. For PUFs, we can restrict our attention to binary vectors $\boldsymbol{x} \in \{0,1\}^{1\times n}$ and the Hamming distance HD therebetween. An attacker knows the probability distribution of $\boldsymbol{x} \leftarrow X$. Consider a noisy version $\widetilde{\boldsymbol{x}}$ of sample $\boldsymbol{x}$. A *secure sketch* [14] is a pair of efficient and possibly randomized procedures: the sketching procedure $\boldsymbol{p} \leftarrow \mathsf{SSGen}(\boldsymbol{x})$, with helper data $\boldsymbol{p} \in \mathcal{P}$, and the recovery procedure $\hat{\boldsymbol{x}} \leftarrow \mathsf{SSRep}(\widetilde{\boldsymbol{x}}, \boldsymbol{p})$. There are two defining properties:

- *Correctness.* If $\mathsf{HD}(\boldsymbol{x}, \widetilde{x}) \leq t$, correctness of reconstruction is guaranteed, i.e., $\hat{\boldsymbol{x}} = \boldsymbol{x}$. If $\mathsf{HD}(\boldsymbol{x}, \widetilde{x}) > t$, there is no guarantee whatsoever.
- *Security.* For a certain lower-bound on the ingoing min-entropy, i.e., $\mathbb{H}_\infty(X) \geq h_{in}$, there is a corresponding lower-bound on the residual min-entropy, i.e., $\widetilde{\mathbb{H}}_\infty(X|P) \geq h_{out}$. 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 $\Delta\mathbb{H}_\infty = \mathbb{H}_\infty(X) - \widetilde{\mathbb{H}}_\infty(X|P)$.

A slightly modified notion brings us to the *fuzzy extractor* [14]. Output $\boldsymbol{k} \in \mathcal{K}$ is then required to be nearly-uniform, given observation $\boldsymbol{p} \leftarrow P$, and is therefore suitable as a secret key. 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., $\boldsymbol{k} \leftarrow \mathsf{Ext}(\boldsymbol{x})$. *Universal hash functions* [9] are good randomness extractors, according to the *(generalized) leftover hash lemma* [16, 3]. Unfortunately, their min-entropy loss is quite substantial. In practice, key generators therefore often rely on a cryptographic hash function that is assumed to behave as a *random oracle*. The latter idealized heuristic results in zero min-entropy loss.

2.5 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 (3) expresses the Hamming bound [23]. 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.

$$\sum_{i=0}^{t} \binom{n}{i} |\mathcal{M}| \leq 2^n. \tag{3}$$

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., $\boldsymbol{w} = \boldsymbol{m} \cdot \boldsymbol{G}$. For any translation $\boldsymbol{\tau} \in \{0,1\}^{1\times n}$ 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. 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 $\mathcal{C}$.

2.6 The Code-Offset Secure Sketch

Several secure sketch constructions rely on a binary code $\mathcal{C}$. For ease of understanding, we focus on the code-offset method of Dodis et al. [14] exclusively. Nevertheless, straightforward equivalencies in App. A prove that all results in this manuscript apply to six other constructions equally well. The code $\mathcal{C}$ that instantiates the code-offset method in Fig. 1 is not necessarily linear. Even more, it is not required be a block code either. Linear codes (BCH, Hamming, repetition, etc.) remain the most frequently used though due to their efficient decoding algorithms [23]. Correctness of reconstruction is guaranteed if $\mathsf{HD}(\boldsymbol{x}, \widetilde{\boldsymbol{x}}) \leq t$, with t the error-correcting capability of the code.

$\boldsymbol{p} \leftarrow \mathsf{SSGen}(\boldsymbol{x})$	$\widehat{\boldsymbol{x}} \leftarrow \mathsf{SSRep}(\widetilde{\boldsymbol{x}}, \boldsymbol{p})$
Random $\boldsymbol{w} \in \mathcal{C}$	$\widetilde{\boldsymbol{w}} \leftarrow \widetilde{\boldsymbol{x}} \oplus \boldsymbol{p} = \boldsymbol{w} \oplus \boldsymbol{e}$
$\boldsymbol{p} \leftarrow \boldsymbol{x} \oplus \boldsymbol{w}$	$\widehat{\boldsymbol{x}} \leftarrow \boldsymbol{p} \oplus \mathsf{Correct}(\widetilde{\boldsymbol{w}})$

Fig. 1. The code-offset secure sketch, having an n-bit reference input $\boldsymbol{x}$.

Min-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 defi-

ciency: $\mathbb{H}_\infty(W) = \log_2(|\mathcal{M}|) < n$. For linear codes in particular, we highlight a convenient interpretation using cosets. Helper data $\boldsymbol{p}$ then reveals in which coset reference $\boldsymbol{x}$ resides. It can be seen easily that $\boldsymbol{p}$ is equal to a random vector in the same coset as $\boldsymbol{x}$. The residual min-entropy in (2) hence reduces to (4) for linear codes, with $\boldsymbol{\epsilon}$ a coset leader. We emphasize that the min-entropy loss $\Delta\mathbb{H}_\infty$ does not depend on the decoding method, simply because the helper data is not affected. For $[n, k, d]$ block codes in particular, the well-known upper bound $\Delta\mathbb{H}_\infty \leq (n-k)$ holds, as proven in [14]. More generally, this extends to $\Delta\mathbb{H}_\infty \leq n - \log_2(|\mathcal{M}|)$.

$$\widetilde{\mathbb{H}}_\infty(X|P) = -\log_2\Big(\mathbb{E}_{\boldsymbol{\epsilon}\leftarrow E}\big[\max_{\boldsymbol{w}\in\mathcal{W}} \mathbb{P}((X = \boldsymbol{\epsilon} \oplus \boldsymbol{w})|(E = \boldsymbol{\epsilon}))\big]\Big). \tag{4}$$

3 Tight Bounds on the Min-Entropy Loss

Currently, secure sketch implementations rely on the $(n-k)$ upper bound on the min-entropy loss, e.g., [28]. Unfortunately, this leads to an overly conservative design when instantiating security parameters accordingly. We develop a graphical framework that produces tight bounds on $\widetilde{\mathbb{H}}_\infty(X|P)$ for typical 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 (5). We also improve upon the rather trivial upper bounds [14] that comprehend the rightmost inequality in (5).

$$\underbrace{\max(\mathbb{H}_\infty(X) - (n - \log_2(|\mathcal{M}|)), 0)}_{\text{worst-case}} \leq \widetilde{\mathbb{H}}_\infty(X|P) \leq \underbrace{\min(\log_2(|\mathcal{M}|), \mathbb{H}_\infty(X))}_{\text{best-case}}. \tag{5}$$

Our lower and upper bounds combined define a relatively narrow interval in which the exact value of $\widetilde{\mathbb{H}}_\infty(X|P)$ is enclosed. We considerably extend related work in [11, 27] 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 [7], full-fledged key generators [28] typically rely on non-trivial codes, e.g., BCH codes [23]. Second, our techniques may be applied to a variety of distributions, while prior work covered biased distributions only. Our bounds remain easy-to-evaluate and are able to handle large codes. Although derived for the code-offset sketch of Dodis et al. [14] in particular, App. A establishes the equivalence with six other constructions.

3.1 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 $\widetilde{\mathbb{H}}_\infty(X|P)$. 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 X for both biased and spatially correlated PUFs. Even for large codes, bounds evaluate in milliseconds-seconds on a standard desktop computer.

- *Biased distribution.* We assume response bits to be independent and identically distributed (i.i.d.) 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. The latter bias 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., *soft-decision decoding* [26, 11] as well as *IBS* [37, 17] and von Neumann [25, 32] debiasing. Therefore, our results enable adequate comparison with related methods, all using a common baseline distribution.
- *Correlated distribution.* We assume response 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.

Fig. 2 specifies the subsets φ_j for both distributions. For the biased distribution, we partition according to $\mathsf{HW}(\boldsymbol{x})$. This corresponds 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 in $\boldsymbol{x}$. 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 [15]. In particular, we separate n indistinguishable stars into j distinguishable bins by adding $j-1$ out of $n-1$ bars.

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 [29], the min-entropy loss of a secure sketch is maximal for a uniformly distributed input, making this a case of special interest.

3.2 Generic Bounds

Equation (6) holds for the code-offset construction of Dodis et al. [14], given that a codeword is selected fully at random during enrollment.

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

Fig. 2. Subsets φ_j for a biased and correlated distribution X, left and right respectively. We define $b_\star = \min(b, 1-b)$ and $c_\star = \min(c, 1-c)$.

$$\mathbb{P}((P=\boldsymbol{p})|(X=\boldsymbol{x})) = \begin{cases} 1/|\mathcal{M}|, & \text{if } \exists \boldsymbol{w} : \boldsymbol{p} = \boldsymbol{x} \oplus \boldsymbol{w} \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Equation (7) applies Bayes' rule to the definition of conditional min-entropy in (2) and fills in (6). 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.

$$\begin{aligned} \widetilde{\mathbb{H}}_\infty(X|P) &= -\log_2\left(\sum_{\boldsymbol{p}\in\mathcal{P}} \cancel{\mathbb{P}(P=\boldsymbol{p})} \max_{\boldsymbol{x}\in\mathcal{X}} \frac{\mathbb{P}(X=\boldsymbol{x})\mathbb{P}((P=\boldsymbol{p})|(X=\boldsymbol{x}))}{\cancel{\mathbb{P}(P=\boldsymbol{p})}}\right) \\ &= -\log_2\left(\frac{1}{|\mathcal{M}|}\sum_{\boldsymbol{p}\in\mathcal{P}} \max_{\boldsymbol{w}\in\mathcal{W}} \mathbb{P}(X=\boldsymbol{p}\oplus\boldsymbol{w})\right). \end{aligned} \tag{7}$$

For linear codes, the workload can be reduced substantially. With a similar derivation as before, we rewrite (4) as shown in (8). Up to 2^n operations suffice. Nevertheless, direct evaluation is only feasible for small codes. We emphasize that our bounds are able to handle large codes, as is typically the case for a practical key generator.

$$\widetilde{\mathbb{H}}_\infty(X|P) = -\log_2\Big(\sum_{\boldsymbol{\epsilon}\in\mathcal{E}} \max_{\boldsymbol{w}\in\mathcal{W}} \mathbb{P}(X=\boldsymbol{\epsilon}\oplus\boldsymbol{w})\Big). \tag{8}$$

Equation (7) iterates over all $\boldsymbol{p}$'s and selects each time the most likely $\boldsymbol{x}$ that is within range, via the addition of a codeword $\boldsymbol{w} \in \mathcal{W}$. We now reverse the roles, as shown in Fig. 3. 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 (7). 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 (9).

$$\widetilde{\mathbb{H}}_\infty(X|P) = -\log_2\left(\frac{1}{|\mathcal{M}|}\sum_{j=1}^{J} s_j^{\boldsymbol{p}} q_j\right). \tag{9}$$

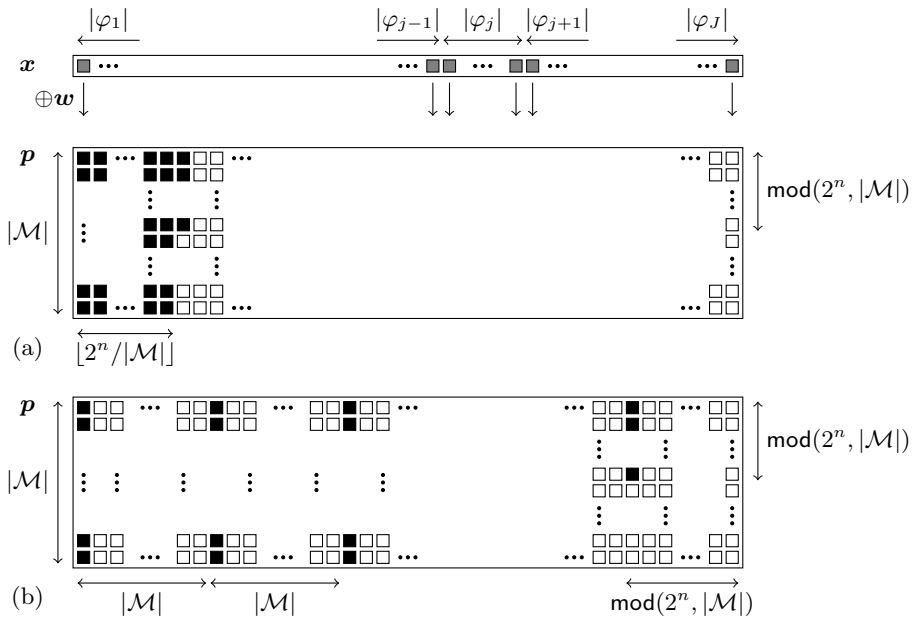


Fig. 3. Reversal of the roles in (7). (a) A lower bound on $\widetilde{\mathbb{H}}_\infty(X|P)$. (b) An upper bound on $\widetilde{\mathbb{H}}_\infty(X|P)$. Black squares represent terms that contribute to $\widetilde{\mathbb{H}}_\infty(X|P)$, one for each $\boldsymbol{p} \in \mathcal{P}$. White squares 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 due to their practical relevance. Fig. 4 swaps the order of iteration in (8). Only one row suffices, i.e., each column of helper data vectors $\boldsymbol{p}$ in Figure 3 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 (10), hereby dropping denominator $|\mathcal{M}|$ compared to (9), given that $s_j^{\boldsymbol{p}} = 2^k \cdot s_j^{\boldsymbol{\epsilon}}$.

$$\widetilde{\mathbb{H}}_\infty(X|P) = -\log_2\left(\sum_{j=1}^{J} s_j^{\boldsymbol{\epsilon}} q_j\right). \tag{10}$$

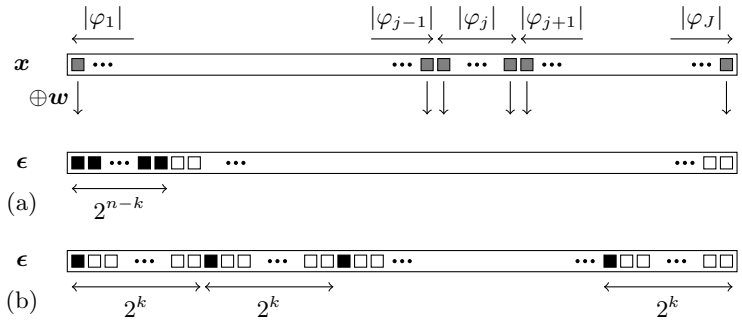


Fig. 4. Reversal of the roles in (8), as applied to linear codes. (a) A lower bound on $\widetilde{\mathbb{H}}_\infty(X|P)$. (b) An upper bound on $\widetilde{\mathbb{H}}_\infty(X|P)$. Black squares represent terms that contribute to $\widetilde{\mathbb{H}}_\infty(X|P)$, one for each $\boldsymbol{\epsilon} \in \mathcal{E}$. 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 $\widetilde{\mathbb{H}}_\infty(X|P)$. 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 $\widetilde{\mathbb{H}}_\infty(X|P)$. 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 Fig. 3 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.

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 $\widetilde{\mathbb{H}}_\infty(X|P) = log_2(|\mathcal{M}|)$, 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 [29] 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.

3.3 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

Algorithm 1: BoundWorstCase

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

Algorithm 2: BoundBestCase

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

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 $\widetilde{\mathbb{H}}_\infty(X|P)$
$j, q, s^{\boldsymbol{p}}, u^{\boldsymbol{p}} \leftarrow 0$
while $s^{\boldsymbol{p}} < 2^n$ **do**
  $j \leftarrow j + 1$
  $u^{\boldsymbol{p}} \leftarrow u^{\boldsymbol{p}} + u_j^{\boldsymbol{p}}$
  $s_j^{\boldsymbol{p}} \leftarrow \min(|\varphi_j||\mathcal{M}|, u^{\boldsymbol{p}} - s^{\boldsymbol{p}})$
  $s_j^{\boldsymbol{p}} \leftarrow \min(s_j^{\boldsymbol{p}}, 2^n - s^{\boldsymbol{p}})$
  $s^{\boldsymbol{p}} \leftarrow s^{\boldsymbol{p}} + s_j^{\boldsymbol{p}}$
  $q \leftarrow q + s_j^{\boldsymbol{p}} \cdot q_j$
$\widetilde{\mathbb{H}}_\infty(X|P) \leftarrow -\log_2(q/|\mathcal{M}|)$

Algorithm 4: BoundBestCase2

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

Worst-Case Bounds We improve the lower bound on $\widetilde{\mathbb{H}}_\infty(X|P)$ 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 [23]. 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 in (11).

$$u_j^{\boldsymbol{p}} = |\mathcal{M}|\frac{|\varphi_j|}{2} = 2^{k-1}|\varphi_j|. \tag{11}$$

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 $\widetilde{\mathbb{H}}_\infty(X|P) = 1$, not depending on parameter c. This is the best-case scenario, given the universal bound $\widetilde{\mathbb{H}}_\infty(X|P) \leq k$. Fig. 5 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 $\{0,1\}^{1\times n}$. As a side note, the result offers yet another [11] refutation of the *repetition code pitfall* of Koeberl et al. [22], a work that ignores that $(n-k)$ is an upper bound only.

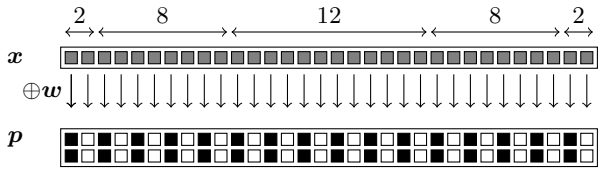


Fig. 5. The exact residual min-entropy $\widetilde{\mathbb{H}}_\infty(X|P)$ for the correlated distribution and an $[n = 5, k = 1, d = 5]$ repetition code.

Best-Case Bounds We improve the upper bound on $\widetilde{\mathbb{H}}_\infty(X|P)$ 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 in (12). Fig. 6 depicts the squares.

$$l_j^{\boldsymbol{p}} = \begin{cases} |\varphi_j||\mathcal{M}|, & \text{if } j \in [1, t+1] \\ 0, & \text{otherwise} \end{cases}. \tag{12}$$

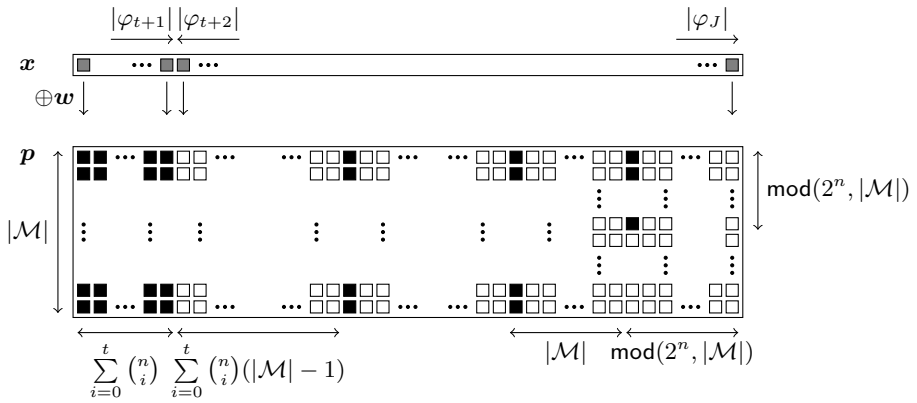


Fig. 6. A tightened upper bound on $\widetilde{\mathbb{H}}_\infty(X|P)$ for the biased distribution, hereby making use of (12).

There is an interesting observation for perfect codes in particular. As clear from the Hamming bound in (3), 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 in (13). This considerably extends a prior result of Delvaux et al. in [11]. The same formula was derived for $[n, k = 1, d = n]$ repetition codes, with n odd. Note that such repetition codes are perfect. The formula was originally adopted to debunk the aforementioned *repetition code pitfall* [22]. Maes et al. [27] fully rebroadcasted the latter contribution from [11] at CHES 2015. It was conveniently unclear though that the novelty thereof is limited to a straightforward conversion from min-entropy to Shannon entropy.

$$\widetilde{\mathbb{H}}_\infty(X|P) = -\log_2\left(\sum_{j=1}^{t+1} |\varphi_j| \cdot q_j\right) = -\log_2(\mathsf{F}_{\mathsf{bino}}(t; n, \min(b, 1-b))). \tag{13}$$

For codes that 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 [10] 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 $b < \frac{1}{2}$, (14) 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 $\widetilde{\mathbb{H}}_\infty(X|P)$ correspondingly.

$$\widetilde{\mathbb{H}}_\infty(X|P) = -\log_2\left(\frac{1}{|\mathcal{M}|}\sum_{h=0}^{h_{\mathsf{cr}}}|\mathcal{E}_h|\cdot|\mathcal{M}|\cdot q_{h+1}\right) = -\log_2\left(\sum_{h=0}^{h_{\mathsf{cr}}}|\mathcal{E}_h|\cdot q_{h+1}\right). \quad (14)$$

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} that 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. Fig. 7 depicts the squares for $n = 4$. Equation (15) evaluates the residual min-entropy.

$$\widetilde{\mathbb{H}}_\infty(X|P) = -\log_2\left(\mathsf{F}_{\mathsf{bino}}(t; n, \min(b, 1-b)) + \frac{1}{2}\binom{n}{\frac{n}{2}}(b(1-b))^{\frac{n}{2}}\right). \quad (15)$$

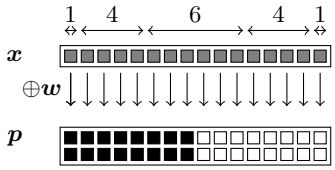


Fig. 7. The exact residual min-entropy $\widetilde{\mathbb{H}}_\infty(X|P)$ for the biased distribution and an $[n = 4, k = 1, d = 4]$ repetition code.

Also for the correlated distribution, distance d might be incorporated to tighten the upper bound on $\widetilde{\mathbb{H}}_\infty(X|P)$. 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 (16). 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 (3).

$$l_j^{\boldsymbol{p}} = \begin{cases} |\mathcal{M}|, & \text{if } j = 1 \\ |\mathcal{M}|\Big(\sum_{h=\lfloor j/2 \rfloor}^{t} \binom{h-1}{\lfloor j/2 \rfloor - 1}\binom{n-h-1}{\lceil j/2 \rceil - 1} & \\ \quad + \sum_{h=\lceil j/2 \rceil}^{t} \binom{h-1}{\lceil j/2 \rceil - 1}\binom{n-h-1}{\lfloor j/2 \rfloor - 1}\Big), & \text{otherwise.} \end{cases} \tag{16}$$

3.4 Numerical Results

Fig. 8 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 (7) and/or (8). As such, the tightness of various bounds can be assessed adequately. Fig. 8(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.

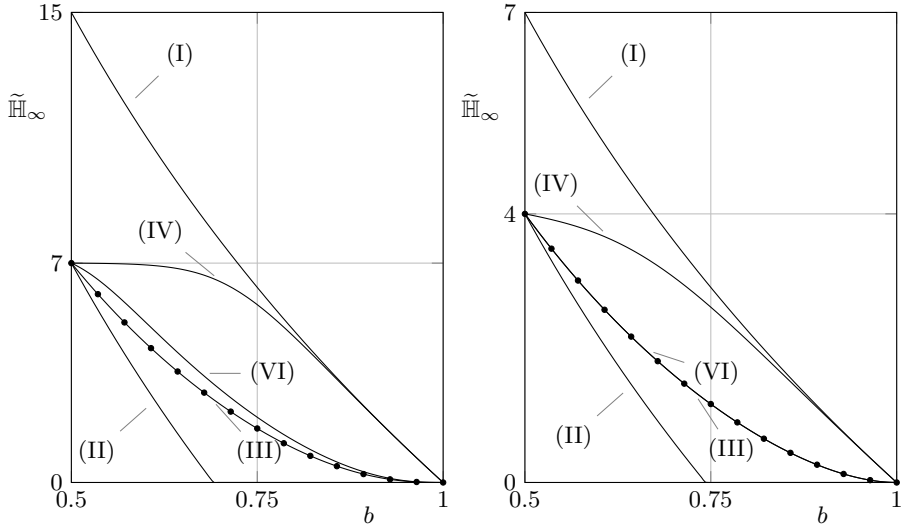
Especially the lower bounds perform well, which benefits a conservative system provider. The best lower bounds in Figs. 8(a), (b) and (c) visually coincide with the exact result. The gap with the $(n - k)$ bound is the most compelling around $b, c \approx 0.7$, where the corresponding curves hit the horizontal axis $\widetilde{\mathbb{H}}_\infty(X|P) = 0$. Also our upper bounds are considerably tighter than their more general alternatives in (5). Nevertheless, the latter bounds remain open for further improvement, with the exception of Fig. 8(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.

4 Applications

The newly developed theory of Section 3 facilitates the design and analysis of error-correction methods, as exemplified in twofold manner. First, we point out a fundamental security flaw in the reverse fuzzy extractor [33]. Second, we provide a motivational framework for debiasing schemes [37, 32, 27].

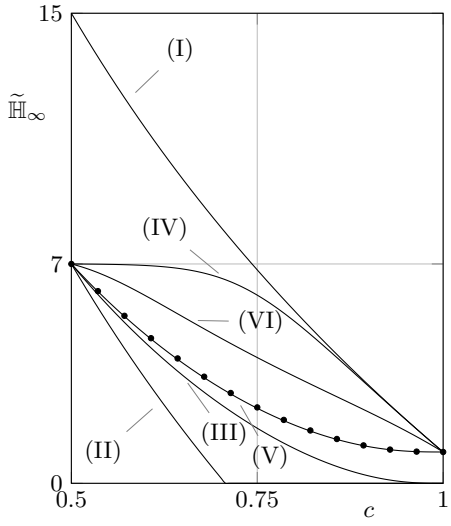
4.1 A Fundamental Security Flaw in Reverse Fuzzy Extractors

The reverse fuzzy extractor, as proposed by Van Herrewege et al. [33] at Financial Crypto 2012, improves the lightweight perspectives of PUF-based authentication protocols. The construction was therefore also adopted in the CHES 2015 protocol of Aysu et al. [2]. Instead of a single helper data exposure only, $\boldsymbol{p} \leftarrow \mathsf{SSGen}(\widetilde{\boldsymbol{x}})$ is regenerated and transferred with each protocol run by a resource-constrained PUF-enabled device. A receiving resource-rich server, storing reference response $\boldsymbol{x}$, can hence reconstruct $\widetilde{\boldsymbol{x}} \leftarrow \mathsf{SSRec}(\boldsymbol{x}, \boldsymbol{p})$ and establish a shared secret as such. The footprint of the device is reduced due to the absence of the heavyweight SSRec procedure.

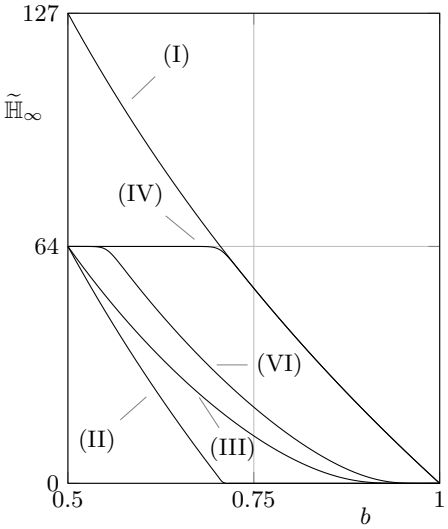


(a) Bias; $[n = 15, k = 7, d = 5]$.

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



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



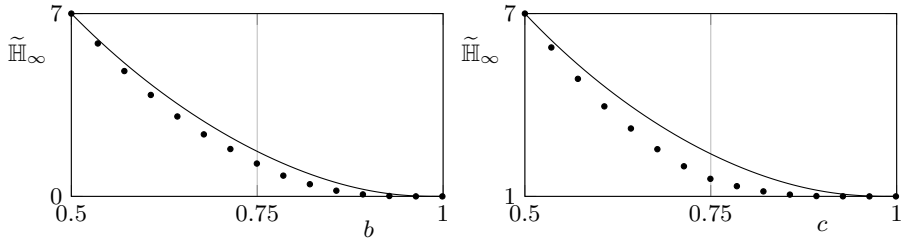
(d) Bias; $[n = 127, k = 64, d = 21]$.

Fig. 8. The secure sketch min-entropy loss for various BCH codes. Dots correspond to an exact exhaustive evaluation of (7)/(8). 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.

We debunk the main security claim that repeated helper data exposure does not result in additional min-entropy loss. The revealed flaw is attributed to the misuse of a reusability proof of Boyen [8]. For the code-offset sketch with linear codes, the exposure of $\boldsymbol{p}_1 \leftarrow \mathsf{SSGen}(\boldsymbol{x})$ and $\boldsymbol{p}_2 \leftarrow \mathsf{SSGen}(\boldsymbol{x} \oplus \boldsymbol{e})$, with perturbation $\boldsymbol{e}$ known and fully determined by the attacker, is provably equivalent. The latter helper data reveals that $\boldsymbol{x}$ belongs to an identical coset $\{\boldsymbol{p}_1 \oplus \boldsymbol{w} : \boldsymbol{w} \in \mathcal{W}\} = \{\boldsymbol{p}_2 \oplus \boldsymbol{e} \oplus \boldsymbol{w} : \boldsymbol{w} \in \mathcal{W}\}$. However, perturbation $\boldsymbol{e}$ is determined by PUF noisiness rather than by the attacker and its release hence reveals new information. Given a sequence of protocol runs, the attacker can approximate all individual bit error rates p_E as well as the coset to which reference $\boldsymbol{x}$ belongs.

Fig. 9 quantifies the residual min-entropy of X with the exclusion and inclusion of revealed bit error rates p_E respectively. In the latter case, we rely on a Monte Carlo evaluation of (17), as enabled by choosing a small $[n = 15, k = 7, d = 5]$ BCH code, given that an analytical approach is not so very straightforward. Exposure of p_E boils down to knowledge of threshold discrepancy $|\boldsymbol{v}(i) - t|$. For the biased distribution, the situation is identical to the flaw in the *soft-decision decoding* scheme of Maes et al. [26]. As pointed out by Delvaux of al. [11], there is a bit-specific bias $b_i = \mathbb{P}(\boldsymbol{r}(i) = 1) = \mathsf{f}_{\mathsf{norm}}(t + |\boldsymbol{v}(i) - t|)/(\mathsf{f}_{\mathsf{norm}}(t+|\boldsymbol{v}(i)-t|)+\mathsf{f}_{\mathsf{norm}}(t-|\boldsymbol{v}(i)-t|))$. For each $\boldsymbol{x}$ in the coset corresponding to $\boldsymbol{p}$, we then compute $\mathbb{P}(X = \boldsymbol{x}) = \prod_{i=1}^{n}(\boldsymbol{x}(i)b_i + (1 - \boldsymbol{x}(i))(1 - b_i))$. Similarly, for the spatially correlated distribution, we compute $\mathbb{P}(X = \boldsymbol{x}) = \mathsf{f}_{\mathsf{norm}}(\boldsymbol{v}, \boldsymbol{0}, \boldsymbol{\Sigma})$, with covariance matrix $\boldsymbol{\Sigma}$ exclusively depending on correlation parameter c, as detailed in App. B.

$$\widetilde{\mathbb{H}}_\infty(X|P) = -\log_2\Big(\mathbb{E}_{\boldsymbol{v}\leftarrow V} \max_{\boldsymbol{w}\in W} \mathbb{P}(V = t + (1 - 2\boldsymbol{w})|\boldsymbol{v} - t| \mid |\boldsymbol{v} - t|\Big). \quad (17)$$



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

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

Fig. 9. The residual min-entropy $\widetilde{\mathbb{H}}_\infty(X|P)$ for a BCH code. The solid lines that exclude revealed bit error rates are computed with BoundWorstCase2; Fig. 8 confirms the visual overlap with the exact result. Dots that include revealed bit error rates correspond to Monte Carlo evaluations of size 10^6.

The revealed flaw differs from existing attacks by Delvaux et al. [12] and Becker [4] that apply to the original protocol [33] exclusively. The latter attacks comprehend the modeling of the highly correlated arbiter PUF via repeated helper data exposure; a preemptive fix can be found in the PhD thesis of Maes [24]. The newly revealed flaw is more fundamentally linked to the reverse fuzzy extractor primitive and applies to all existing protocols so far [33, 24, 2]. Observe in Fig. 9 that the overly conservative $(n-k)$ bound would compensate for the additional unanticipated min-entropy loss. However, this somewhat defeats the purpose in light of the original lightweight intentions, and this observation might not necessarily hold for every possible distribution. Further theoretical work may determine to which extent and at which cost reverse fuzzy extractors can be repaired. A potential fix already exists for biased distributions, as illustrated later-on.

4.2 Motivation for Debiasing Schemes

Debiasing schemes transform a biased PUF-induced distribution into a uniform distribution. A considerable fraction of the response bits is discarded in order to restore the balance between 0 and 1. Indices of retained bits are stored as helper data. A subsequent secure sketch, known to have an exact min-entropy loss of $(n-k)$ bits for uniform inputs, still corrects the errors. A first debiasing proposal is the *index-based syndrome* (IBS) scheme of Yu et al. [37], further generalized by Hiller et al. [17]. Second, there are the von Neumann debiasing schemes, as presented by Maes et al. [27] at CHES 2015. The latter authors advertised to have solved an *important open problem*, hereby conveniently overlooking the existence of IBS. Another convenient oversight is that the basic von Neumann debiasing scheme had already been proposed in the PhD thesis of Van Herrewege [32].

Prior debiasing schemes assume that a stand-alone sketch cannot handle biased distributions. This conclusion originates from either the extrapolation of repetition code insights and/or application of the $(n-k)$ bound. The precise entropy loss behavior for larger codes, e.g., a BCH $[n=127, k=64, d=21]$ code as in Fig. 8, was an educated guess so far. Our newly derived bounds clearly resolve this motivational uncertainty, in addition to making stand-alone sketches more competitive. For low-bias situations, e.g., $b \in [0.4, 0.6]$, the $(n-k)$ already resulted in a competitive sketch [27]; the new bounds can only improve hereupon. We emphasize that modern high-quality PUFs tend to have a low bias. Notable cases of a high bias can typically be attributed to an avoidable asymmetry in the circuit. Nevertheless, for high-bias situations, the new bounds clearly indicate the need of debiasing schemes. The benefit is amplified by choosing a sketch with a k-bit output, several of which are listed in App. A. The uniform output is then directly usable as a key, hereby eliminating the Hash function and its additional min-entropy loss in case the leftover hash lemma is applied.

Finally, we highlight that one of the von Neumann debiasing schemes in [27] was claimed to be reusable. Surprisingly, this claim holds, despite explicitly overlooking the misuse of Boyen's proof once again and stating that a stand-alone sketch is reusable. A lucky side effect of retaining pairs of alternating bits

only, i.e., 01 and 10, is that the imbalance in error rates between 0 and 1 cannot be observed in the helper data. The scheme is considerably less efficient than other von Neumann variants though, showing that reusability comes at a price.

5 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 established the first significant stepping stone. New techniques may have to be developed in order to tackle more advanced *second-order* distributions. Elaborating a wider range of applications would be another area of progress. We hope to have showcased the potential by debunking the main security claim of the reverse fuzzy extractor and by providing proper quantitative motivation for debiasing schemes.

Acknowledgment

The authors greatly appreciate the support received. The European Union's Horizon 2020 research and innovation programme under grant number 644052 (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. A. Aysu, E. Gulcan, D. Moriyama, P. Schaumont, and M. Yung. End-To-End Design of a PUF-Based Privacy Preserving Authentication Protocol. In *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, September 13-16, 2015, Proceedings*, pages 556–576, 2015.
3. 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.

4. G. T. Becker. On the Pitfalls of Using Arbiter-PUFs as Building Blocks. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 34(8):1295–1307, 2015.
5. 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.
6. M. Bhargava and K. Mai. An efficient reliable PUF-based cryptographic key generator in 65nm CMOS. In *Design, Automation & Test in Europe Conference & Exhibition, DATE 2014, Dresden, Germany, March 24-28, 2014*, pages 1–6, 2014.
7. 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.
8. X. Boyen. Reusable cryptographic fuzzy extractors. In *Proceedings of the 11th ACM Conference on Computer and Communications Security, CCS 2004, Washington, DC, USA, October 25-29, 2004*, pages 82–91, 2004.
9. L. Carter and M. N. Wegman. Universal Classes of Hash Functions. *Journal of Computer and System Sciences*, 18(2):143–154, 1979.
10. 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.
11. 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.
12. J. Delvaux, R. Peeters, D. Gu, and I. Verbauwhede. A Survey on Lightweight Entity Authentication with Strong PUFs. *ACM Comput. Surv.*, 48(2):26, 2015.
13. J. Delvaux and I. Verbauwhede. Fault Injection Modeling Attacks on 65 nm Arbiter and RO Sum PUFs via Environmental Changes. *IEEE Trans. on Circuits and Systems*, 61-I(6):1701–1713, 2014.
14. 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.
15. W. Feller. *An Introduction to Probability Theory and Its Applications, Vol. 1, 3rd Edition.* 1968.
16. 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.
17. M. Hiller, D. Merli, F. Stumpf, and G. Sigl. Complementary IBS: application specific error correction for PUFs. In *2012 IEEE International Symposium on Hardware-Oriented Security and Trust, HOST 2012, June 3-4, 2012*, pages 1–6, 2012.
18. 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.
19. 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.
20. A. Juels and M. Wattenberg. A Fuzzy Commitment Scheme. In *CCS 1999, 6th ACM Conference on Computer and Communications Security*, pages 28–36, 1999.
21. 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.

22. P. Koeberl, J. Li, A. Rajan, and W. Wu. Entropy loss in PUF-based key generation schemes: The repetition code pitfall. In *2014 IEEE International Symposium on Hardware-Oriented Security and Trust, HOST 2014, Arlington, VA, USA, May 6-7, 2014*, pages 44–49, 2014.
23. F. J. MacWiliams and N. J. A. Sloane. *The theory of error correcting codes*. 1977.
24. R. Maes. *Physically Unclonable Functions: Constructions, Properties and Applications*. PhD thesis, KU Leuven, 2012. Ingrid Verbauwhede (promotor).
25. R. Maes. An accurate probabilistic reliability model for silicon PUFs. In *Cryptographic Hardware and Embedded Systems - CHES 2013 - 15th International Workshop, Santa Barbara, CA, USA, August 20-23, 2013. Proceedings*, pages 73–89, 2013.
26. 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.
27. R. Maes, V. van der Leest, E. van der Sluis, and F. Willems. Secure key generation from biased PUFs. In *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, September 13-16, 2015, Proceedings*, pages 517–534, 2015.
28. 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.
29. L. Reyzin. Entropy Loss is Maximal for Uniform Inputs. Technical Report BUCS-TR-2007-011, Department of Computer Science, Boston University, September 2007.
30. 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.
31. 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.
32. A. Van Herrewege. *Lightweight PUF-based Key and Random Number Generation*. PhD thesis, KU Leuven, 2015. Ingrid Verbauwhede (promotor).
33. A. Van Herrewege, S. Katzenbeisser, R. Maes, R. Peeters, A. Sadeghi, I. Verbauwhede, and C. Wachsmann. Reverse Fuzzy Extractors: Enabling Lightweight Mutual Authentication for PUF-Enabled RFIDs. In *Financial Cryptography and Data Security - 16th International Conference, FC 2012, Kralendijk, Bonaire, Februray 27-March 2, 2012, Revised Selected Papers*, pages 374–389, 2012.
34. 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.
35. 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.
36. 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.
37. 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 Secure Sketch Equivalency Proofs

Bounds previously derived for the code-offset method of Dodis et al. [14] apply to six other constructions equally well. For convenience, we generalize the original secure sketch so that its reconstructed output $\boldsymbol{y} \leftarrow \mathsf{SSRep}(\boldsymbol{x}, \boldsymbol{p})$ is not necessarily equal to $\boldsymbol{x}$. As such, the prior notion of *fuzzy commitment* [20] can be supported as well. Hereby, we commit to a secret value $\boldsymbol{y}$ by binding it to $\boldsymbol{x}$. One may decommit given an $\widetilde{\boldsymbol{x}}$ that is sufficiently close to $\boldsymbol{x}$. Constructions that return a substring of $\boldsymbol{x}$, e.g., [21], are supported too. The fuzzy extractor definition offers intrinsic support for both cases, without any modifications from our part. The key is still computed as $\boldsymbol{k} \leftarrow \mathsf{Hash}(\boldsymbol{y})$.

Fig. 10 specifies the seven secure sketch constructions of interest, all instantiated with a binary code $\mathcal{C}$. We now review additional coding theory, before transitioning to individual sketch discussions. 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, $\boldsymbol{H} = (\boldsymbol{A}^T \| \boldsymbol{I}_{n-k})$. There is a one-to-one correspondence between cosets and syndromes [23].

All seven constructions exhibit an identical min-entropy loss. Or more precisely, all have the same residual min-entropy $\mathbb{H}_\infty(Y|P)$ given in (18), as long as the ingoing distribution X and the code $\mathcal{C}$ are identical. A consequence thereof is that the well-known $(n-k)$ upper bound on the min-entropy loss as well as our newly derived bounds apply to all seven sketches. Simple equivalency proofs are established in pairwise manner, as guided by Fig. 11. Several pairwise equivalencies were already established in existing literature, e.g., [34, 11], but these often impose unnecessary restrictions on the distribution. We hence make progress in terms of completeness and generality.

$$\widetilde{\mathbb{H}}_\infty(Y|P) = -\log_2\Big(\mathbb{E}_{\boldsymbol{p}\leftarrow P}\big[\max_{\boldsymbol{y}\in\mathcal{Y}} \mathbb{P}((Y=\boldsymbol{y})|(P=\boldsymbol{p}))\big]\Big). \tag{18}$$

A.1 Code-Offset Methods of Juels et al., Dodis et al. and Tuyls et al.

The code-offset method of Juels et al. [20] is represented by Fig. 10(a). The code $\mathcal{C}$ is not necessarily linear. Even more, it is not required be a block code either. Fig. 10(b) represents a modification where Rep returns sketch input $\boldsymbol{x}$ rather than codeword $\boldsymbol{w}$, as proposed by Dodis et al. [14]. 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. Fig. 10(c) represents another minor modification where Rep returns message $\boldsymbol{m}$, as suggested by Tuyls et al. [30]. This necessitates an implementation of Decode rather than $\mathsf{Correct}$.

$\boldsymbol{p} \leftarrow \mathsf{SSGen}(\boldsymbol{x})$	$\widehat{\boldsymbol{y}} \leftarrow \mathsf{SSRep}(\widetilde{\boldsymbol{x}}, \boldsymbol{p})$	
Random $\boldsymbol{w} \in \mathcal{C}$ $\boldsymbol{p} \leftarrow \boldsymbol{x} \oplus \boldsymbol{w}$	$\widetilde{\boldsymbol{w}} \leftarrow \widetilde{\boldsymbol{x}} \oplus \boldsymbol{p} = \boldsymbol{w} \oplus \boldsymbol{e}$ $\widehat{\boldsymbol{y}} = \widehat{\boldsymbol{w}} \leftarrow \mathsf{Correct}(\widetilde{\boldsymbol{w}})$	(a) Code-offset method of Juels et al. [20].
	$\widetilde{\boldsymbol{w}} \leftarrow \widetilde{\boldsymbol{x}} \oplus \boldsymbol{p} = \boldsymbol{w} \oplus \boldsymbol{e}$ $\widehat{\boldsymbol{y}} = \widehat{\boldsymbol{x}} \leftarrow \boldsymbol{p} \oplus \mathsf{Correct}(\widetilde{\boldsymbol{w}})$	(b) Code-offset method of Dodis et al. [14].
	$\widetilde{\boldsymbol{w}} \leftarrow \widetilde{\boldsymbol{x}} \oplus \boldsymbol{p} = \boldsymbol{w} \oplus \boldsymbol{e}$ $\widehat{\boldsymbol{y}} = \widehat{\boldsymbol{m}} \leftarrow \mathsf{Decode}(\widetilde{\boldsymbol{w}})$	(c) Code-offset method of Tuyls et al. [30].
$\boldsymbol{p} \leftarrow \boldsymbol{x} \cdot \boldsymbol{H}^T$	$\boldsymbol{s} \leftarrow \widetilde{\boldsymbol{x}} \cdot \boldsymbol{H}^T \oplus \boldsymbol{p} = \boldsymbol{e} \cdot \boldsymbol{H}^T$ Determine $\widehat{\boldsymbol{e}}$ $\widehat{\boldsymbol{y}} = \widehat{\boldsymbol{x}} \leftarrow \widetilde{\boldsymbol{x}} \oplus \widehat{\boldsymbol{e}}$	(d) Syndrome method of Bennett et al. [5].
$\boldsymbol{p} \leftarrow \boldsymbol{x}(1:k) \cdot \boldsymbol{A}$ $\oplus\ \boldsymbol{x}(k+1:n)$	$\widehat{\boldsymbol{w}} \leftarrow \mathsf{Correct}(\widetilde{\boldsymbol{x}} \oplus (\boldsymbol{0} \| \boldsymbol{p}))$ $\widehat{\boldsymbol{y}} = \widehat{\boldsymbol{x}} \leftarrow \widehat{\boldsymbol{w}} \oplus (\boldsymbol{0} \| \boldsymbol{p})$	(e) Systematic method of Yu [36].
	$\widehat{\boldsymbol{y}} = \widehat{\boldsymbol{x}}(1:k) \leftarrow \mathsf{Decode}(\widetilde{\boldsymbol{x}}$ $\oplus(\boldsymbol{0} \| \boldsymbol{p}))$	(f) Systematic method of Kang et al. [21].
$\boldsymbol{p} \leftarrow j$ 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. 10. Seven secure sketch constructions, all having an n-bit input $\boldsymbol{x}$. Correctness of reconstruction is guaranteed, given a noisy version $\widetilde{\boldsymbol{x}}$ with $\mathsf{HD}(\boldsymbol{x}, \widetilde{\boldsymbol{x}}) \leq t$.

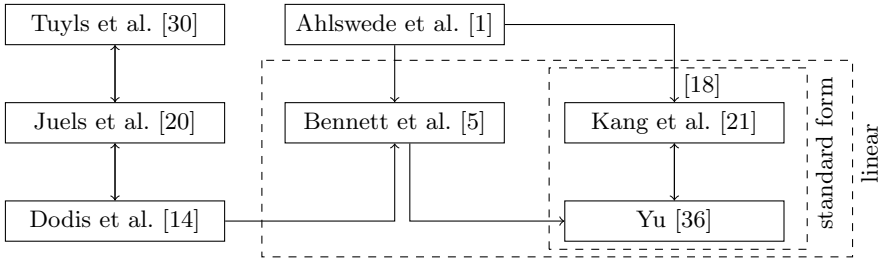


Fig. 11. Pairwise min-entropy loss equivalencies among seven sketches, as indicated by the arrows. Transitive relations apply when following the arrows. E.g., the schemes of Dodis et al. and Kang et al. are equivalent, given that both are instantiated with a linear code in standard form.

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 in (19). 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, (18) evaluates to the same value for all three methods. Note that $|\mathcal{M}| = |\mathcal{W}| = |\mathcal{X}'|$.

$$\begin{aligned}\forall(\boldsymbol{p},\boldsymbol{m}) \in (\mathcal{P}\times\mathcal{M}), \mathbb{P}((M=\boldsymbol{m})|(P=\boldsymbol{p})) &= \mathbb{P}((W=\mathsf{Encode}(\boldsymbol{m}))|(P=\boldsymbol{p})) \\ &= \mathbb{P}((X=\mathsf{Encode}(\boldsymbol{m})\oplus\boldsymbol{p})|(P=\boldsymbol{p})).\end{aligned} \tag{19}$$

A.2 Syndrome Method of Bennett et al.

The syndrome method of Bennett et al. [5] is represented by Fig. 10(d). Although initially proposed as part of a *quantum oblivious transfer* protocol, it maps quite easily to the secure sketch framework of Dodis et al. [14]. 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. [14]. This is a trivial consequence from the universally valid expression in (20), given that the helper data $\boldsymbol{p}$ is limited to $(n-k)$ bits.

$$\widetilde{\mathbb{H}}_\infty(X|P) \geq \widetilde{\mathbb{H}}_\infty(X) - \log_2(|\mathcal{P}|). \tag{20}$$

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. The residual min-entropy of both methods can hence be written as shown in (4).

A.3 Systematic Methods of Yu and Kang et al.

The method of Yu [36] is represented by Fig. 10(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 (20), given that helper data $\boldsymbol{p}$ is limited to $(n-k)$ bits. Fig. 10(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 [21] and independently also by Hiller et al. in [18]. Nevertheless, (21) indicates that the residual min-entropy is identical. The main insight is that $\boldsymbol{x}(1:k)$ and $\boldsymbol{p}$ fully determine $\boldsymbol{x}(k+1:n)$.

$$\begin{aligned}\forall(\boldsymbol{p},i,\boldsymbol{x}) \in (\mathcal{P}\times\mathcal{X}), \mathbb{P}((X(1:k)=\boldsymbol{x}(1:k))|((P=\boldsymbol{p}))) \\ = \mathbb{P}((X=(\boldsymbol{x}(1:k)\|(\boldsymbol{x}(1:k)\cdot\boldsymbol{A}\oplus\boldsymbol{p}))|((P=\boldsymbol{p}))).\end{aligned} \tag{21}$$

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 (22). 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.

$$\boldsymbol{p} = \boldsymbol{x} \cdot \boldsymbol{H}^T = \boldsymbol{x} \cdot \begin{pmatrix} \boldsymbol{A} \\ \boldsymbol{I}_{n-k} \end{pmatrix} = \boldsymbol{x}(1:k) \cdot \boldsymbol{A} \oplus \ \boldsymbol{x}(k+1:n). \tag{22}$$

A.4 Multi-Code Method of Ahlswede et al.

The method of Ahlswede et al. [1] is represented by Fig. 10(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. [18]. 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 (20), given that helper data $\boldsymbol{p} = j$ is limited to $(n-k)$ bits.

In [18], 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 Fig. 10(f), i.e., helper data $\boldsymbol{p}$ and reconstructed output $\boldsymbol{y}$ are identical. We consider a slightly more general case. In particular, a linear code $\mathcal{C}_0$ that 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 (4) holds once again. The one-to-one correspondence of output $\boldsymbol{y}$ in (23) finalizes our proof.

$$\forall(\boldsymbol{p}, \boldsymbol{x}) \in (\mathcal{P} \times \mathcal{X}), \mathbb{P}((X = \boldsymbol{x}) | (P = \boldsymbol{p})) = \mathbb{P}((M = \mathsf{Decode}_{\mathcal{C}_{\boldsymbol{p}}}(\boldsymbol{x})) | (P = \boldsymbol{p})). \tag{23}$$

A.5 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 [7]. 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 $n = z \cdot n_2 \cdot \frac{n_1}{k_2}$ and $k = z \cdot k_1$. Therefore, prior equivalencies still apply.

B Covariance Matrix $\boldsymbol{\Sigma}$ for the Correlated Distribution

We determine the covariance matrix $\boldsymbol{\Sigma}$ for a spatially correlated distribution X with parameter c. The probability c_{ij} that response bits $\boldsymbol{x}(i)$ and $\boldsymbol{x}(j)$ are equal

is given in (24). For $i = j$ and $|i - j| = 1$, the expression reduces to 1 and c respectively.

$$c_{ij} = \mathbb{P}(\boldsymbol{x}(i) = \boldsymbol{x}(j)) = \sum_{u=0}^{\lfloor |i-j|/2 \rfloor} \mathsf{f}_{\mathsf{bino}}(2u; |i-j|, 1-c), \quad \text{with } i, j \in [1, n]. \tag{24}$$

For $i = j$, the variance $\boldsymbol{\Sigma}(i, i) = \mathbb{E}[(\boldsymbol{v}(i))^2] = 1$. For $i \neq j$, a link between covariance $\boldsymbol{\Sigma}(i, j) = \mathbb{E}[\boldsymbol{v}(i)\boldsymbol{v}(j)]$ and c_{ij} is established in (25).

$$c_{ij} = 2 \int_0^\infty \int_0^\infty \mathsf{f}_{\mathsf{norm}}\left(\begin{pmatrix} v_i & v_j \end{pmatrix}; \begin{pmatrix} 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & \boldsymbol{\Sigma}(i,j) \\ \boldsymbol{\Sigma}(i,j) & 1 \end{pmatrix} \right) \mathrm{d}v_i \mathrm{d}v_j. \tag{25}$$

Integration in polar coordinates results in the more convenient relation in (26).

$$\boldsymbol{\Sigma}(i, j) = \sin\Big(\pi\Big(c_{ij} - \frac{1}{2}\Big)\Big). \tag{26}$$