An efficient fuzzy extractor for limited noise

Boris Škorić and Pim Tuyls

Abstract

A fuzzy extractor is a security primitive that allows for reproducible extraction of an almost uniform key from a noisy non-uniform source. We analyze a fuzzy extractor scheme that uses universal hash functions for both information reconciliation and privacy amplification. This is a useful scheme when the number of error patterns likely to occur is limited, regardless of the error probabilities. We derive a sharp bound on the uniformity of the extracted key, making use of the concatenation property of universal hash functions and a recent tight formulation of the leftover hash lemma. We show that using almost universal hash functions instead of universal leads only to a small penalty in the number of extracted key bits, while giving a large reduction of the storage requirements.

1 Introduction

1.1 Security with noisy data

Many security applications require input bitstrings to be uniformly distributed and exactly reproducible. Cryptographic keys, for instance, have to be uniformly random in order to prevent attacks; they have to be reproducible in order to allow for decryption of encrypted data, verification of signatures, successful authentication etc. Even a single bit error in a key causes failure. Physical sources of randomness, however, are neither uniform nor noise-free. The patterns in biometrics such as fingerprints and iris scans do not follow a uniform distribution, and they are never exactly reproduced when a measurement is repeated. Measurement noise can be due to many factors, e.g. differences in lighting conditions or sensor alignment, physiological changes, difference between sensors etc. Another class of physical sources that has received a lot of attention recently are the Physical Unclonable Functions (PUFs), also known as Physical One-Way Functions, Physical Random Functions and Physically Obscured Keys. PUFs can be regarded as 'non-biological biometrics'. Many types of PUF have been described in the literature, e.g. multiple scattering of laser light [15], reflection of laser light from paper fibers [2], randomized dielectrics in protective chip coatings [22], radiofrequent responses from pieces of metal [7] or thin-film resonators [24], delay times in chip components [5] and start-up values of SRAM cells [11].

For security and/or privacy reasons it is often necessary to apply a one-way hash function to the biometric/PUF measurement, in analogy with the `/etc/passwd` file in UNIX. The storage of biometric/PUF data is assumed to be public; the hashing step hides the measurement data. However, as measurements are noisy, it is not possible to directly hash; a single bit error in the input causes roughly 50% of the output bits to flip. Hence, an error-correction step is required first ('information reconciliation'). This is not trivial, since the redundancy data has to be stored publicly and may reveal too much sensitive information. Similarly, if PUF data is to be used as a key, then it should be thoroughly noise-corrected first. Here, too, it is crucial that the publicly stored redundancy data does not reveal secrets.

After information reconciliation, the step of *privacy amplification* is applied, mapping a non-uniform random string to a shorter, almost uniform string. The requirement of uniformity is obvious in the case of key extraction. Interestingly, extracting uniform bitstrings is also desirable in biometric systems and PUF-based anti-counterfeiting, applications where the identifiers are *not* considered to be secret. A uniform string is the most efficient way of storing the entropy present in a measurement. Furthermore, database search speed is improved.

The concept of a *Fuzzy Extractor* [9, 10], also known as a *helper data scheme* [14], was introduced as a primitive that achieves both information reconciliation and privacy amplification. The publicly stored enrolment data (a.k.a. secure sketch, helper data or public data) suffices to reproducibly reconstruct a string from noisy measurements, yet leaks only a negligible amount of information about the extracted key. An overview of privacy-preserving biometrics, PUFs and fuzzy extraction is given in [23].

1.2 Problems with noise correction

One of the nontrivial aspects of the information reconciliation step is the 'shape' of the noise. The noise patterns are not always nicely compatible with a representation in terms of binary strings. Error-correcting codes (ECCs) work best on (binary) strings under the condition that the likely to occur error patterns are completely random. This is the case e.g. for i.i.d. bit errors and for burst errors that have no preference for a specific location in the bit string. Now consider an N-dimensional biometric feature vector (or PUF output) being the source. Such a source is typically not binary. Mapping the feature vector to a binary string introduces problems for standard ECCs in the following ways:

- It often happens that the errors are not uniformly random, e.g. certain burst errors are far more likely than others.
- It is also common for error probabilities to depend on the value of the feature vector itself.
- Often, one-dimensional components of the feature vector are separately discretized [22], and the discretization intervals are assigned a binary representation such as a Gray code. This procedure causes unequal error probabilities of the bits that form the Gray code. (One bit flips when the noise nudges the value one interval to the left, another one flips when the noise nudges it one interval to the right; all the other bits have very low bit error probabilities.) Furthermore, the bit error probabilities depend on the value of the feature vector.
- When several components of the feature vector are combined into a D-dimensional space, the binarization sometimes leads to asymmetries in the bit representation of equally likely errors. For instance, when a two-dimensional space is discretized according to a hexagonal lattice [3], and the noise is random, then the noise will nudge the feature value (center of a hexagon) to one of the surrounding hexagons with equal probability, but the number of bit flips is not the same for these six errors.

Even under these circumstances, an ECC is capable of dealing with errors no matter what their probability distribution is. But there is a price to pay: The number of redundancy bits in the code is far higher than what an 'ideal' code would have. If X and X' are two different measurements of the source, then an ideal code would be able to extract $I(X; X')$ bits of information. (Here I denotes the mutual information.) All the asymmetries listed above reduce the entropy of the error patterns and hence increase the mutual entropy $I(X; X')$. Typical ECCs are not able to capitalize on the low entropy of the errors, since they must be able to correct the 'worst case' errors, and consequently a large part of the entropy present in the source gets wasted. Furthermore, ECCs typically approach the Shannon bound only when the code words are very long.
The challenge is to construct a practical error correction method that, in the case of very non-uniform noise probabilities, extracts more information than typical ECCs.

1.3 Related work

A lot of work has been done to convert data structures with various error patterns into binary representations that allow for the use of error-correcting codes. (See e.g. [10] for an overview of schemes for Hamming distance, set difference and edit distance). In this paper we follow a different approach. We restrict ourselves to the case where the noise is in a certain sense well-behaved:

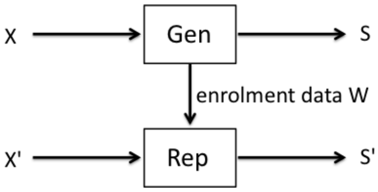

Figure 1: *Fuzzy Extractor.* `Gen` *generates public data P and a near-uniform key S.* `Rep` *tries to reproduce S from P and a noisy measurement X'.*

the error patterns may be very bad, and the noise may be very strong, but the number of error patterns that are likely to occur is limited.[1]

The information reconciliation problem for PUFs and biometrics can be seen as a special case of the Slepian-Wolf problem [18] with a single encoder and a single decoder. Fig. 1 shows the two main procedures in a Fuzzy Extractor: The `Gen` procedure extracts a key S from the source X and generates public data P; in the Slepian-Wolf setting `Gen` is the encoder and P would be called 'side information'. The `Rep` procedure reproduces S from P and a noisy measurement X'. In the Slepian-Wolf setting this corresponds to the decoder. A generic solution in this setting is *Slepian-Wolf coding* [18]. It amounts to creating a codebook of random codewords for the typical set. Given X', receiving such a codeword is sufficient to determine which of the candidates X, jointly typical with X', was enrolled, provided that the codeword has entropy of at least $\mathsf{H}(X|X')$. In this paper we consider the case where the size of the codebook is 'manageable'.

One approach to implement Slepian-Wolf coding efficiently is to use *Universal Hash Functions* (UHFs) [4] or a slight relaxation thereof, *Almost Universal Hash Functions* (AUHFs) [20]. These functions are efficient to compute and behave like perfectly random functions as far as collisions in the target space are concerned. Their use for Slepian-Wolf coding is known [19, 10].

A Fuzzy Extractor has to achieve more than just error correction. First, P must not leak too much about S. Second, S has to be as close to uniform as possible (privacy amplification). For general sources, uniformity can be achieved by using (A)UHFs.[2] Thus, we see that (A)UHFs provide an efficient way to achieve information reconciliation as well as privacy amplification when the source is ill behaved.

Another aspect of Fuzzy Extractors is so-called 'robustness'. This refers to the property that the `Gen` procedure is able to detect whether $S' = S$, and output an error message when $S' \neq S$. In this way adversarial modifications of the public data and/or the PUF can be detected. (It is assumed that the attacker does not know the secret S.) In the absence of a public key infrastructure, robustness is achieved basically by using the derived key itself to create an authentication code over the error correction data. This code becomes part of the public data. The first such construction was given in [1], and requires the random oracle assumption. Dodis et al. [8] presented a construction in the standard model, for certain distance metrics, which, in the case of zero noise, works whenever the min-entropy rate of the source exceeds 1/2. The sources we consider, however, typically have a much lower entropy rate. Furthermore we will not make any assumptions about the existence of a metric. For these reasons we will not be able to use constructions like [8].

A better robust fuzzy extractor construction was given in [6] for the Common Reference String (CRS) model. It needs no assumptions about the entropy rate. The robustness is based on message authentication codes with Key Manipulation Security (KMS-MACs).

[1] An example of such a source is a two-dimensional subspace of a noisy biometric, discretized to a hexagonal lattice [3]. Errors occur with fairly high probability (X and X' do not map to the same hexagon), but the number of ways in which this can occur is limited.

[2] For a source X with a lot of known structure in its probability distribution, using a compression algorithm may be feasible [13]; then the extracted entropy is close to the Shannon entropy of X, which is much better than what is achieved by universal hashing (see Section 2). However, our aim is a fuzzy extractor for sources X whose statistics are not so well known.

1.4 Contributions in this paper

We analyze an offline fuzzy extractor scheme that employs (A)UHFs *for both privacy amplification and information reconciliation.* By 'offline' we mean that communication between Alice and Bob is only one-way. A first hash function is applied to X to create a short string that serves as helper data. It is just long enough to allow for reconstruction of X from X'. The secret key is extracted by applying a second hash function to X. Such a scheme has several advantages:

- Information reconciliation is efficient even if the errors are highly non-uniform and strongly correlated with the data, as long as the likely number of possible error patterns is limited.
- Computation of a short almost universal hash can be done efficiently. Hence it is feasible to compute a large number of hashes.
- The hardware cost for implementation of the fuzzy extractor is very modest. Furthermore, by shrinking the footprint of the error correction circuit, the number of possible hardware attack points is reduced.

Two concatenated AUHFs together form a new AUHF. This property is useful for security proofs. We derive a sharp bound on the uniformity of the extracted key, given that the attacker sees the public data. We make use of the concatenation property of AUHFs and a recent tighter formulation of the leftover hash lemma [25]. The error correction data and the extracted key are considered to be part of the same big hash value. If this is taken literally, then it can be said that the scheme performs information reconciliation and privacy amplification *at the same time* or even *in the opposite order* compared to other schemes.

We formulate our main result as a choice of key length $c(\varepsilon)$ such that the distance of the key's distribution from uniformity is upper bounded by ε. Use of the leftover hash lemma yields an expression for $c(\varepsilon)$ consisting of two parts: a positive term depending on the source entropy and a negative 'penalty' term which becomes more severe with decreasing ε. Revealing k bits of the big hash as helper data has two effects on $c(\varepsilon)$. (i) a trivial reduction of the key length by k bits; (ii) nontrivial correction terms in the penalty term, arising from the fact that the key and the helper data are derived from the same source.

We assume that the device that reconstructs the key is too computationally constrained to perform asymmetric crypto operations such as signature verification. Furthermore, we assume that the channel over which the public data is communicated from Alice to Bob is neither tamper-proof nor tamper-evident. Hence, authentication of the helper data is possible only by using a *robust* fuzzy extractor. Since we do not wish to rely on the random oracle assumption, we adopt the KMS-MAC based construction of [6], which only requires the existence of a Common Reference String (CRS).

We argue that this type of KMS-MAC is in fact too strong for our purposes, since the attacker has less knowledge than assumed in [6]. In particular, we show that our fuzzy extractor scheme does not have the 'linearity' property.

Finally we show that the use of AUHFs instead of UHFs carries a small penalty in terms of extracted bits, but has a large benefit for the size of the random extractor seeds. This is important for storage-constrained devices.

2 Preliminaries

Random variables are denoted in capitals. Sets are denoted in calligraphic font (e.g. $X \in \mathcal{X}$). For $X, X' \in \mathcal{X}$, we define the statistical distance as

$$\Delta(X; X') = \tfrac{1}{2} \sum_{x \in \mathcal{X}} \Big| \text{Prob}[X = x] - \text{Prob}[X' = x] \Big| .$$

We do not use any notion of distance between X and X' in feature vector space. We use a very general approach to model the measurement noise.

Definition 1 *Let $\theta \in (0,1)$ be a fixed parameter. Let $X \in \mathcal{X}$ be the enrolment measurement and $X' \in \mathcal{X}$ be the verification measurement. A set $B \subset \mathcal{X}$ is called an* **incoming $(1-\theta)$-neighborhood** *of x' if*

$$\sum_{x \in B} \mathrm{Prob}[X = x | X' = x'] \geq 1 - \theta. \tag{1}$$

The set of all incoming $(1-\theta)$-neighborhoods of x' is denoted as $\mathcal{B}^{\mathrm{in}}_{1-\theta}(x')$.

We assume that the probability distribution of the (possibly X-dependent) *noise* is known sufficiently accurately to allow for explicit construction of $(1-\theta)$-neighborhoods. Example: $\mathcal{X} = \Sigma^n$, where Σ is a finite alphabet, with the Hamming distance between X and X' bounded by some constant, and an accurately known symbol error probability.

Definition 2 *Let $\eta > 0$. Let $\mathcal{R}$, $\mathcal{X}$ and $\mathcal{Z}$ be finite sets. Let $\{\Phi_r\}_{r \in \mathcal{R}}$ be a family of hash functions from $\mathcal{X}$ to $\mathcal{Z}$. The family $\{\Phi_r\}_{r \in \mathcal{R}}$ is called η-***almost universal** *iff, for R drawn uniformly from $\mathcal{R}$, it holds that*

$$\mathrm{Prob}[\Phi_R(x) = \Phi_R(x')] \leq \eta$$

for all $x, x' \in \mathcal{X}$ with $x' \neq x$. In the special case $\eta = 1/|\mathcal{Z}|$ the family is called **universal***.*

Lemma 1 *Let $\{\Phi_r\}_{r \in \mathcal{R}} : \mathcal{X} \to \{0,1\}^\ell$ be a $2^{-\ell}(1+\delta_\Phi)$-almost universal family of hash functions. Let $\{\Psi_t\}_{t \in \mathcal{T}} : \mathcal{X} \to \{0,1\}^k$ be a $2^{-k}(1+\delta_\Psi)$-almost universal family of hash functions. Then the concatenation $\{\Psi_t || \Phi_r\}_{t \in \mathcal{T}, r \in \mathcal{R}}$ is an $2^{-k-\ell}(1+\delta_\Psi)(1+\delta_\Phi)$-almost universal family of hash functions from $\mathcal{X}$ to $\{0,1\}^{k+\ell}$.*

The Leftover Hash Lemma dictates how many near-uniform key bits Alice and Bob can extract from X if they apply (A)UHFs. In its most tight formulation, the lemma involves a quantity called *smooth Rényi entropy*. Below we briefly review the definition of this entropy measure and show the Leftover Hash Lemma.

Definition 3 *(Paraphrased from [12].) Let $\mathbb{P}$ be a probability measure on $\mathcal{X}$. Let $\rho \geq 0$. We define the* **strictly bounded ρ-vicinity** *of $\mathbb{P}$ as*

$$B^\rho(\mathbb{P}) = \left\{ \mathbb{Q} : \forall_{x \in \mathcal{X}}\ \mathbb{Q}(x) \leq \mathbb{P}(x) \text{ and } \sum_{x \in \mathcal{X}} \mathbb{Q}(x) \geq 1 - \rho \right\}.$$

Definition 4 *Let $\mathbb{P}$ be a probability measure on $\mathcal{X}$. Let $\rho \geq 0$. The* **smooth Rényi entropy** *of $\mathbb{P}$ is*

$$\mathsf{H}^\rho_\alpha(\mathbb{P}) = \max_{\mathbb{Q} \in B^\rho(\mathbb{P})} \mathsf{H}_\alpha(\mathbb{Q}).$$

Here $\mathsf{H}_\alpha(\mathbb{Q})$ denotes the ordinary Rényi entropy $\frac{-1}{\alpha-1} \log \sum_x [\mathbb{Q}(x)]^\alpha$.

Definition 5 *Let $X \in \mathcal{X}$ be a random variable. Let $R \in \mathcal{R}$ be a uniformly distributed random variable, independent of X. For any $\varepsilon > 0$ we say that a finite set $\mathcal{Z}$ is ε-allowed if there exists a function $F : \mathcal{X} \times \mathcal{R} \to \mathcal{Z}$ such that $\Delta(RF(X,R);\ RU) \leq \varepsilon$, where U is a random variable uniformly distributed on $\mathcal{Z}$, independent of X. The ε-***extractable randomness** *of X is defined as*

$$\ell^\varepsilon_{\mathrm{ext}}(X) = \max \{\log |\mathcal{Z}| : \mathcal{Z} \textit{ is } \varepsilon\textit{-allowed}\}.$$

Lemma 2 *(From [25]; tighter version of the result in [16].) Let $\varepsilon \geq 0$. Let X be a random variable on $\mathcal{X}$. Let $\{\Phi_r\}_{r \in \mathcal{R}}$ be an η-almost universal family of hash functions from $\mathcal{X}$ to $\mathcal{T}$, with $\eta = (1+\delta)/|\mathcal{T}|$. Then the ε-extractable randomness from X using this family of hash functions is bounded from below by*

$$\max_{\rho \in [0, \varepsilon - \delta/[4\varepsilon])} \left[\mathsf{H}^\rho_2(X) + 2 - \log \frac{1}{\varepsilon(\varepsilon - \rho) - \delta/4} \right]. \tag{2}$$

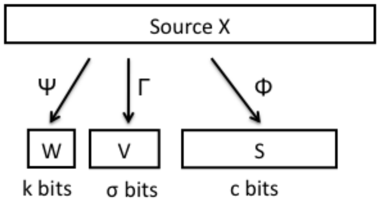

Figure 2: *The AUHFs Ψ_t, Γ_j and Φ_r compresses X to k, σ and c bits, respectively. The concatenation $\Psi\Gamma\Phi$ is also an AUHF.*

3 Offline key reconstruction

We present a scheme for offline key reconstruction, i.e. with only one-way communication. The two parties, called Alice and Bob, are for instance a device manufacturer and a PUF device, or a biometric enrollment authority and a biometric authentication system. The scheme is depicted in Fig. 3.

3.1 Offline key reconstruction protocol

System setup phase:
Alice and Bob beforehand agree on three almost universal families of hash functions $\{\Phi_r\}_{r\in\mathcal{R}} : \mathcal{X} \to \{0,1\}^c$, $\{\Psi_t\}_{t\in\mathcal{T}} : \mathcal{X} \to \{0,1\}^k$ and $\{\Gamma_j\}_{j\in\mathcal{J}} : \mathcal{X} \to \{0,1\}^\sigma$. (See Fig. 2.) These are $2^{-c}(1+\delta_\Phi)$, $2^{-k}(1+\delta_\Psi)$ and $2^{-\sigma}(1+\delta_\Gamma)$ almost universal, respectively. The Common Reference String (CRS) consists of the random values $\{r,t,j\}$. Alice and Bob also agree on a function $F : \{0,1\}^\sigma \times \{0,1\}^* \to \{0,1\}^m$ that uses a σ-bit key to produce an m-bit authentication code. The Φ, Ψ, Γ hash families are known to the attacker, as are c, σ, k, F and the CRS.

Enrolment phase:

1. Alice performs a measurement and obtains an outcome x.
2. She computes $s = \Phi_r(x)$, $w = \Psi_t(x)$, $v = \Gamma_j(x)$ and $a = F(v,w)$.
3. She stores w and a in nonvolatile memory.

Attack phase:
The attacker modifies $\{w,a\}$ to $\{\tilde{w},\tilde{a}\}$.

Reconstruction phase:

1. Bob reads $\{\tilde{w},\tilde{a}\}$ from the nonvolatile memory and $\{r,t,j\}$ from the CRS.
2. Bob performs a measurement and obtains an outcome x'. He chooses a neighborhood $B \in \mathcal{B}^{\rm in}_{1-\theta}(x')$. He compiles a list $L = \{x_i \in B : \Psi_t(x_i) = \tilde{w}\}$. If $L = \emptyset$, the protocol aborts in failure.
3. For all $x_i \in L$, Bob computes $v_i := \Gamma_j(x_i)$. He checks if $F(v_i,\tilde{w}) = \tilde{a}$. In the event that a single match x^* occurs, the protocol is considered to have succeeded, and Bob accepts $\tilde{s} := \Phi_r(x^*)$ as the reconstructed shared secret. If there are no matches, or more than one, then the protocol aborts in failure.

Remarks:
(i) In Bob's step 3, the event $L = \emptyset$ occurs with probability at most θ.
(ii) In Bob's step 4, the verification of a serves to authenticate the helper data w. See the discussion of robustness in Section 3.2.2.

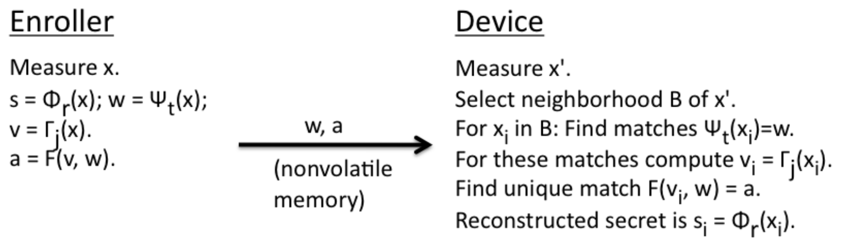
Enroller
Measure x.
$s = \Phi_r(x)$; $w = \Psi_t(x)$;
$v = \Gamma_j(x)$.
$a = F(v, w)$.
w, a
(nonvolatile memory)
Device
Measure x'.
Select neighborhood B of x'.
For x_i in B: Find matches $\Psi_t(x_i)$=w.
For these matches compute $v_i = \Gamma_j(x_i)$.
Find unique match $F(v_i, w) = a$.
Reconstructed secret is $s_i = \Phi_r(x_i)$.

Figure 3: *The offline key reconstruction scheme in the CRS model.*

(iii) The parameter k must be chosen sufficiently large, so that Bob does not have to compute too many v_i values in step 4. The expected number of elements in L is of order $|B|2^{-k}$. The requirement of having the correct $F(v_i, w)$ further restricts the number of candidates to $|B|2^{-k-m}$. Hence, in order to reduce the probability of multiple matches in Bob's step 4 below some constant γ, we need $k + m = \mathcal{O}(\log|B| + \log 1/\gamma)$.

3.2 Security analysis of the offline key reconstruction

3.2.1 Uniformity of the key S

An eavesdropping attacker, Eve, has access to t, r, j w, a. The first part of the security analysis concerns passive attackers, and amounts to determining the effect of Eve's knowledge on the security of the key s. As a security measure we use the statistical distance from the uniform distribution. We have the following result.

Theorem 1 *Consider the protocol of Section 3.1. Let* $\delta = (1 + \delta_\Psi)(1 + \delta_\Phi)(1 + \delta_\Gamma) - 1$. *If* c, k, σ *satisfy*

$$c \leq \max_\rho \left[\mathsf{H}_2^\rho(X) + 2 - \log\frac{1}{\varepsilon(\varepsilon - \rho) - \delta/4}\right] - k - \sigma \tag{3}$$

then

$$\Delta(RTJWA\,S;\; RTJWA\,U_c) \leq \varepsilon,$$

where U_c *is a random variable uniformly distributed on* $\{0, 1\}^c$, *independent of* X, R, T *and* J.

The theorem states that, averaged over all R, T, J, W, A, the distribution of the key S, given Eve's knowledge, is ε-close to uniform. I.e. the inequality can be formulated as

$$\mathbb{E}_{rtjwa}\left[\Delta\left(S|R = r, T = t, J = j, W = w, A = a;\; U_c\right)\right] \leq \varepsilon,$$

where $\mathbb{E}$ stands for the expectation value.
Proof: A is a function of R, T, J, W, V, hence the combined variable $RTJWA$ is a function of the combined variable $RTJWV$. We use the fact that applying a function cannot increase the statistical distance. Thus

$$\Delta(RTJWAS; RTJWAU_c) \leq \Delta(RTJWVS; RTJWVU_c).$$

Next, for any random variables $X \in \mathcal{X}, Y \in \mathcal{Y}$ it holds that $\Delta(XY; U_\mathcal{X}Y) \leq \Delta(XY; U_{\mathcal{X}\times\mathcal{Y}})$, where $U_\mathcal{X}$ is a variable uniform on $\mathcal{X}$. This gives

$$\Delta(RTJWVS; RTJWVU_c) \leq \Delta(RTJWVS; RTJU_{k+\sigma+c}).$$

According to Lemma 1 the concatenation WVS is a $2^{-k-\sigma-c}(1 + \delta)$-almost universal hash, with $1 + \delta = (1 + \delta_\Psi)(1 + \delta_\Gamma)(1 + \delta_\Phi)$. Finally we apply Lemma 2 to the hash WVS to find how big $k + \sigma + c$ can be while still having WVS ε-close to uniformity. □

The result (3) has a simple form. The ε-extractable randomness from X is given by the "$\max_\rho$" expression. Revealing k bits of helper data reduces the entropy of S by at most k bits. Employing σ bits of extracted randomness as an authentication key uses up (at most) a further σ bits of the entropy of S.

However, Eq.(3) is not trivial. The parameter δ does not only depend on the choice of Φ, but also on the choice of the functions Ψ and Γ. This happens because the distribution of S, *conditioned on W and V*, becomes less uniform when W and V become less uniform. We see from (3) that all three parameters δ_Ψ, δ_Γ, δ_Φ have to be significantly smaller than ε^2, otherwise they cause a loss of extractable entropy.

3.2.2 Active attacks

The second part of the security analysis considers an active attacker who has one of the following objectives: (i) denial of service (DoS), or (ii) tricking Bob into accepting a false key $\tilde{s} \neq s$. We assume that the attacker does not know x, x' or s.

DoS is easy to achieve. Damaging the PUF and/or randomly modifying the public data $\{w, a\}$ results in protocol abortion with overwhelming probability. In case of a DoS attack, Bob will not be able to distinguish between such an attack and an occurrence of exceptionally strong noise.

Tricking Bob into accepting a false $\tilde{s} \neq s$ is much more difficult than DoS. The attacker has to concoct fake values $\{\tilde{w}, \tilde{a}\}$ such that there exists a point $\hat{x}$ in the neighborhood of x' satisfying $\Psi_t(\hat{x}) = \tilde{w}$, *and* $\tilde{a} = F(\Gamma_j(\hat{x}),\ \tilde{w})$. These tasks are made difficult by the fact that the attacker does not know x, x', v. For ordinary MACs F the 'difficulty' is hard to determine quantitatively; their security is defined under the assumption that the MAC key is not modified by the attacker. Therefore we let F be a MAC with key manipulation security (KMS-MAC) [6].

Definition 6 *(From [6]). An (M, G, T, γ)-KMS-MAC is a function $F : \mathcal{G} \times \mathcal{M} \to \mathcal{T}$ which maps a source message in a set $\mathcal{M}$ of size M to a tag in a finite set $\mathcal{T}$ of size T using a key from a group $\mathcal{G}$ of order G. The function F satisfies, for any $\tilde{\mu} \neq \mu \in \mathcal{M}$, any $\tau, \tilde{\tau} \in \mathcal{T}$, and any $\Delta \in \mathcal{G}$,*

$$\mathrm{Prob}\left[F(K + \Delta, \tilde{\mu}) = \tilde{\tau} \mid F(K, \mu) = \tau\right] \leq \gamma,$$

where the probability is taken over a uniformly random key $K \in \mathcal{G}$.

According to this definition, in the context of fuzzy extractors, the probability of successful forgery is bounded even if the attacker knows how his choice of $\tilde{\mu}$ (the fake helper data) affects the extracted key: Cramer et al. adopt a special 'linearity' property such that, although K itself is hidden from the attacker, the key offset Δ is a known function of μ and $\tilde{\mu}$. They also gave a KMS-MAC construction with performance (tag and key length) close to ordinary MACs.

We argue that in the case of our AUHF-based helper data, Def. 6 allows unrealistically strong attacks. The attacker's task is to find some $\tilde{w} \neq w$ with the property that $\tilde{w} = \Psi_t(\hat{x})$ for some $\hat{x}$ close to x. (If he fails, then Bob gets $L = \emptyset$ in step 2 of the reconstruction phase.) He has to guess, since he has no knowledge of x. Now consider a parameter choice such that $p := |B| 2^{-k} < 1$. Then for randomly chosen $\tilde{w} \neq w$ there is a probability $1 - p$ of Bob getting $L = \emptyset$. This already breaks the 'linearity' property of [6]. Furthermore, even for $p > 1$ linearity does not hold. Now the chances are overwhelming that there exists a proper $\hat{x}$ close to x for any value $\tilde{w} \neq w$. However, the attacker does not know $\hat{x}$; he only has a list $\Psi_t^{\mathrm{inv}}(\tilde{w})$ of candidates,

$$\Psi_t^{\mathrm{inv}}(\tilde{w}) := \{y \in \mathcal{X} \mid \Psi_t(y) = \tilde{w}\}.$$

The possible MAC keys that can result are $\Gamma_j(\Psi_t^{\mathrm{inv}}(\tilde{w}))$, where Γ may be a completely different function than Ψ. In general, when Ψ and Γ are AUHFs there is no structure present that would cause the list of possible offsets $\Delta = \Gamma_j(\Psi_t^{\mathrm{inv}}(\tilde{w})) - \Gamma_j(x)$ to become a single value. Hence the linearity property does not hold.[3]

[3] There are other examples of helper data without the linearity property. Consider for instance helper data in the form of reliable component selection [21]. Part of the helper data w specifies which parts of X are very likely to be noise-free (reliable components) and hence should be used for key extraction. The attacker then cannot predict changes in the key resulting from modifications in the list of components.

We conclude that for our helper data scheme it is prudent to adopt a MAC with some protection against key manipulation, but not necessarily the kind proposed in [6], since it protects against stronger-than-realistic attacks.

3.2.3 Doing without the Common Reference String

It is possible to modify the scheme so that there is no CRS. Then the hashing seeds $\{r, t, j\}$ become part of the public data stored in nonvolatile memory. In the attack phase the seeds may be modified by the attacker. Hence the tag a has to cover not only w, but also $\{r, t, j\}$ so that all public data are MAC'ed. The MAC has to be a KMS-MAC as proposed in [6], since it has to be resistant against manipulation of j, which causes *linear* changes in the MAC key $v = \Gamma_j(x)$.

4 Implementation issues

As mentioned, our scheme is only practical if Bob's $(1-\theta)$-neighborhood of x' is not too large. A second important point is the implementation of the hash functions Ψ, Φ, Γ. The Ψ hash is especially critical, since it has to be run on the whole $(1-\theta)$-neighborhood of x'. Fortunately, efficient implementations are known. The 'PR' and 'WH' universal hashes proposed in [26], for instance, only need operations in $\mathrm{GF}(2^k)$, which are well suited for low-power hardware. Furthermore, it is useful to split up Ψ, e.g. into b-bit sub-hashes: this allows Bob to check the first b bits of $\Psi_t(x_i)$ against the first b bits of w, already reducing the number of candidate x_i by a factor 2^{-b} before having to compute the rest of the hash. Each subsequent sub-hash achieves another factor 2^{-b}.
Another important implementation aspect is the length of the (public) random strings r, t and j. They have to be stored, and on constrained devices there is often a limit to the amount of memory. Let us consider the Ψ family. Typical constructions of a universal family of hash functions require that $\log|\mathcal{T}|$ is (almost) as large as $\log|\mathcal{X}|$. For instance, the construction of Example 8.39 in [17] requires #bits $= \log|\mathcal{T}| = \log|\mathcal{X}| - k$. For highly non-uniform sources X this is prohibitive. It is possible to save on memory by relaxing the constraints on the hash function: By allowing almost-universality (Def. 2), one gets a tradeoff between the quality of the privacy amplification and the space needed to store t. There are constructions (see e.g. [17]) of $(1+\delta_\Psi)2^{-k}$-almost universal hash functions that require only

$$\log|\mathcal{T}| = \mathcal{O}\left(k - \log k + \log\log|\mathcal{X}| + \log[1/\delta_\Psi]\right). \tag{4}$$

We see that the dependence on $|\mathcal{X}|$ has changed from $\log|\mathcal{X}|$ to $\log\log|\mathcal{X}|$, which is much smaller. Hence, when storage is constrained it may pay off to use an almost-universal instead of a perfectly universal hash function.
In the CRS-less scenario there is a second benefit of the reduced size of $\{r, t, j\}$: The length of the authentication key v can be reduced, leaving more entropy for the shared secret s.

Acknowledgements

We kindly thank Stefan Katzenbeisser and an anonymous reviewer for useful comments.

References

[1] X. Boyen. Reusable cryptographic fuzzy extractors. In *ACM CCS 2004*, pages 82–91.

[2] J. D. R Buchanan, R. P. Cowburn, A. Jausovec, D. Petit, P. Seem, G. Xiong, D. Atkinson, K. Fenton, D.A. Allwood, and M.T. Bryan. Forgery: 'fingerprinting' documents and packaging. *Nature, Brief Communications*, 436:475, July 2005.

[3] I. Buhan, J. Doumen, P. Hartel, and R. Veldhuis. Constructing practical fuzzy extractors using QIM. Technical Report TR-CTIT-07-52, 2007. http://doc.utwente.nl/59974/.

[4] J.L. Carter and M.N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18(2):143–154, 1979.

[5] D. Clarke, B. Gassend, M. van Dijk, and S. Devadas. Secure hardware processors using silicon physical one-way functions. In *ACM CCS 2002*.

[6] R. Cramer, Y. Dodis, S. Fehr, C. Padró, and D. Wichs. Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors. In N.P. Smart, editor, *EUROCRYPT*, volume 4965 of *Lecture Notes in Computer Science*, pages 471–488. Springer, 2008.

[7] G. DeJean and D. Kirovski. Radio frequency certificates of authenticity. In *IEEE Antenna and Propagation Symposium – URSI*, 2006.

[8] Y. Dodis, J. Katz, L. Reyzin, and A. Smith. Robust fuzzy extractors and authenticated key agreement from close secrets. In C. Dwork, editor, *CRYPTO*, volume 4117 of *Lecture Notes in Computer Science*, pages 232–250. Springer, 2006.

[9] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. Cryptology ePrint Archive, Report 2003/235.

[10] 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.

[11] J. Guajardo, S.S. Kumar, G.J. Schrijen, and P. Tuyls. FPGA intrinsic PUFs and their use for IP protection. In *CHES 2007*, volume 4727 of *LNCS*, pages 63–80.

[12] T. Holenstein and R. Renner. On the randomness of independent experiments, 2006. `http://arxiv.org/abs/cs.IT/0608007`.

[13] T. Ignatenko, G.J. Schrijen, B. Škorić, P. Tuyls, and F.M.J. Willems. Estimating the secrecy rate of physical uncloneable functions with the context-tree weighting method. In *ISIT 2006*, pages 499–503.

[14] J.-P. M. G. Linnartz and P. Tuyls. New shielding functions to enhance privacy and prevent misuse of biometric templates. In *Conference on Audio and Video Based Person Authentication*, volume 2688 of *LNCS*, pages 238–250, 2003.

[15] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld. Physical one-way functions. *Science*, 297:2026–2030, Sept. 2002.

[16] R. Renner and S. Wolf. Simple and tight bounds for information reconciliation and privacy amplification. In *ASIACRYPT 2005*, volume 3788 of *LNCS*, pages 199–216.

[17] V. Shoup. *A Computational Introduction to Number Theory and Algebra.* Cambridge University Press, 2008. version 2; `http://www.shoup.net/ntb/ntb-v2.pdf`.

[18] D. Slepian and J. Wolf. Noiseless coding of correlated information sources. *IEEE Trans. on Inf. Theory*, 19(4):471–480, 1973.

[19] A. Smith. Scrambling adversarial errors using few random bits, optimal information reconciliation, and better private codes. In *SODA 2007*, pages 395–404.

[20] D.R. Stinson. Universal hashing and authentication codes. *Designs, Codes, and Cryptography*, 4:369–380, 1994.

[21] 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 T. Kanade, A.K. Jain, and N.K. Ratha, editors, *AVBPA*, volume 3546 of *Lecture Notes in Computer Science*, pages 436–446. Springer, 2005.

[22] P. Tuyls, G.J. Schrijen, B. Škorić, J. van Geloven, R. Verhaegh, and R. Wolters. Read-proof hardware from protective coatings. In *CHES 2006*, volume 4249 of *LNCS*, pages 369–383.

[23] P. Tuyls, B. Škorić, and T. Kevenaar. *Security with Noisy Data: Private Biometrics, Secure Key Storage and Anti-Counterfeiting*. Springer, 2007.

[24] B. Škorić, T. Bel, A.H.M. Blom, B.R. de Jong, H. Kretschman, and A.J.M. Nellissen. Randomized resonators as uniquely identifiable anti-counterfeiting tags. Secure Component and System Identification (SECSI) Workshop, March 2008.

[25] B. Škorić, C. Obi, E. Verbitskiy, and B. Schoenmakers. Sharp lower bounds on the extractable randomness from non-uniform sources, 2008. `http://eprint.iacr.org/2008/484`.

[26] K. Yüksel, J.-P. Kaps, and B. Sunar. Universal hash functions for emerging ultra-low-power networks. In *CNDS 2004*.