Fuzzy Extractors*

A Brief Survey of Results from 2004 to 2006

Yevgeniy Dodis[†] Leonid Reyzin[‡] Adam Smith[§]

April 9, 2008

1 Motivation

This survey presents a general approach for handling secret biometric data in cryptographic applications. The generality manifests itself in two ways: we attempt to minimize the assumptions we make about the data, and to present techniques that are broadly applicable wherever biometric inputs are used.

Because biometric data comes from a variety of sources that are mostly outside of anyone's control, it is prudent to assume as little as possible about how they are distributed; in particular, an adversary may know more about a distribution than a system's designers and users. Of course, one may attempt to measure some properties of a biometric distribution, but relying on such measurements in the security analysis is dangerous, because the adversary may have even more accurate measurements available to it. For instance, even assuming that some property of a biometric behaves according to a binomial distribution (or some similar discretization of the normal distribution), one could determine the mean of this distribution only to within $\approx \frac{1}{\sqrt{n}}$ after taking n samples; a well-motivated adversary can take more measurements, and thus determine the mean more accurately.

Rather than assuming that some statistical information about the biometric input is available, we assume only that the input is unpredictable: i.e., that if an adversary is allowed a single guess at the value of the input, the likelihood that it is correct is 2^{-m} for some m. This is a minimal assumption in the applications we consider: indeed, if the input is easily guessed, then one cannot use it to derive, say, a secret key for encryption or remote authentication. Of course, determining the exact value of m may in itself present a challenge; however, some lower bound on m is necessary for any sort of security claims.

Similarly, while some understanding of errors in biometric measurements is possible, we prefer to minimize the assumptions we make about such errors. We assume only that a subsequent measurement is within a given, allowed distance of the measurement taken at enrollment.

The broad applicability of the approaches presented here stems from the initial observation that many prior solutions for specific security problems based on noisy data (including biometrics)

---

*A prior version of this article appears as Chapter 5 of [43]

[†]New York University, http://cs.nyu.edu/~dodis/. Supported by the National Science Foundation grants CCR-0133806 and CCR-0311095, and by the New York University Research Challenge Fund 25-74100-N5237.

[‡]Boston University, http://www.cs.bu.edu/~reyzin. Supported by National Science Foundation grants CCF-0515100 and CNS-0546614, and the Institute for Pure and Applied Mathematics at UCLA.

[§]Pennsylvania State University, http://www.cse.psu.edu/~asmith. This survey was written while L.R. and A.S. were visiting the Institute for Pure and Applies Mathematics at UCLA.

shared essential techniques and analyses. Instead of designing solutions for each particular setting as it arises, it seems worthwhile to consider the properties that such solutions share, and encapsulate them into primitives that can be used in a variety of contexts.

There is a large body of cryptographic literature that can provide security if only there is a secret, uniformly random, reliably reproducible random string (such a string can be used, for example, as a secret key, or as a seed to generate a public/private key pair). Therefore, if biometric inputs can be converted to such strings, a wide array of cryptographic techniques can be used to provide security from biometrics. To do so, we present a primitive termed *fuzzy extractor*. It extracts a uniformly random string R from its input w in a noise-tolerant way: if the input changes to some w' but remains close, the string R can be reproduced exactly. To help in the reproduction of R, a fuzzy extractor, when used for the first time, outputs a helper string P that can safely be made public without decreasing the security of R.

Fuzzy extractors can be used, for example, to encrypt and authenticate a user's record using his biometric input as a key: one uses R as an encryption/authentication key, and stores P in order to recover R from the biometric whenever the record needs to be accessed. Note that R is not stored—thus, the user's biometric itself effectively acts as the key, and the record can be stored encrypted in a nonsecret, even replicated, database with the confidence that only when the correct biometric is presented will the record be decrypted.

As a step in constructing fuzzy extractors, and as an interesting object in its own right, we present another primitive, termed *secure sketch*. It allows precise reconstruction of a noisy input, as follows: on input w, a procedure outputs a sketch s. Then, given s and a value w' close to w, it is possible to recover w. The sketch is secure in the sense it does not reveal much about w: w retains much of its entropy even if s is known. Thus, instead of storing w for fear that later readings will be noisy, it is possible to store s instead, while retaining much of the secrecy of w.

Because different biometric information has different error patterns, we do not assume any particular notion of closeness between w' and w. Rather, in defining our primitives, we simply assume that w comes from some metric space, and that w' is no more than a certain distance t from w in that space. We consider particular metrics only when building concrete constructions, which we provide for the Hamming distance, set difference and edit distance (these are defined in Section 3). Constructions for other notions of distance are possible as well, and some have appeared in the literature, e.g. [10, 28]. Of course, biometric measurements often have to be processed before they fall into a convenient metric space; for instance, techniques such as IrisCode [14] convert images of irises into strings in the Hamming space. This processing is nontrivial and is itself an active research area; in particular, Chapter 16 of [43] discusses transformations of biometric measurements (feature vectors) into binary strings.

We also present several extensions of secure sketches and fuzzy extractors. In Section 4 we consider slight relaxations of the reliability requirement (reducing it from perfect reliability to reliability with high probability or against computationally bounded adversaries), which permit constructions with significantly higher error tolerance. In Section 5 we consider strengthening the privacy requirement, so that not only does w remain unpredictable to an adversary who has s or P, but also any adversarially-chosen function of w (such as, for instance, ethnic origin, gender, or diseases revealed by the biometric) remains as hard to compute as without s or P. In Section 6 we strengthen fuzzy extractors to remain secure even in the face of active adversaries who may modify P; in particular, this allows for key agreement between a user and a server based only on a biometric input.

The techniques presented have applications beyond biometrics to other settings where noisy inputs are used, such as inputs from human memory, drawings used as passwords, keys from quantum channels or noisy channels, etc. They share common roots with prior work in these settings, as well as with work on communication-efficient information reconciliation. One unexpected application of fuzzy extractors is the proof of impossibility of certain strong notions of privacy for statistical databases [23].

2 Basic Definitions

2.1 Predictability, Min-Entropy, Statistical Distance, Extraction

If X is a random variable, we will also denote by X the probability distribution on the range of the variable. We use U_ℓ to denote the uniform distribution $\{0,1\}^\ell$. Two occurrences of the same random variable in a given expression mean that the same value is used in both (rather than two independent samples). If an algorithm (or a function) f is randomized, we denote by $f(x;r)$ the result of computing f on input x with randomness r.

For security, one is often interested in the probability that the adversary predicts a random value (e.g., guesses a secret key). The adversary's best strategy is to guess the most likely value. Thus, the *predictability* of a random variable A is $\max_a \mathbb{P}[A = a]$, and the *min-entropy* $\mathsf{H}_\infty(A)$ is $-\log(\max_a \mathbb{P}[A = a])$ (min-entropy can thus be viewed as the "worst-case" entropy [11]). A random variable with min-entropy at least m is called an m-source.

Consider now a pair of (possibly correlated) random variables A, B. If the adversary finds out the value b of B, then the predictability of A becomes $\max_a \mathbb{P}[A = a \mid B = b]$. On average, the adversary's chance of success in predicting A is the $\mathbb{E}_{b\leftarrow B}[\max_a \mathbb{P}[A = a \mid B = b]]$. Here we are taking the *average* over B (which is not under adversarial control), but the *worst case* over A (because prediction of A is adversarial once b is known). Again, it is convenient to talk about security in log-scale, which is why *conditional* min-entropy of A given B is defined in [18, Section 2.4] as simply the logarithm of the above:

$$\widetilde{\mathsf{H}}_\infty(A \mid B) \stackrel{\text{def}}{=} -\log \mathop{\mathbb{E}}_{b\leftarrow B}\left[\max_a \mathbb{P}[A = a|B = b]\right] = -\log \mathop{\mathbb{E}}_{b\leftarrow B}\left[2^{-\mathsf{H}_\infty(A|B=b)}\right]$$

This measure satisfies a weak chain rule, namely, revealing any λ bits of information about A can cause its entropy to drop by at most λ [18, Lemma 2.2]: if $B \in \{0,1\}^\lambda$ then $\widetilde{\mathsf{H}}_\infty(A \mid B, C) \geq \widetilde{\mathsf{H}}_\infty(A \mid C) - \lambda$. This definition of conditional min-entropy is suitable for cryptographic purposes and, in particular, for extracting nearly uniform randomness from A (see [18, Appendix B] for a discussion of related entropy measures, such as smooth entropy [36]). The notion of "nearly" here corresponds to the *statistical distance* between two probability distributions A and B, defined as $\mathsf{SD}[A, B] = \frac{1}{2}\sum_v |\mathbb{P}[A = v] - \mathbb{P}[B = v]|$. We can interpret this as a measure of distinguishability: any system in which B is replaced by A will behave exactly the same as the original, with probability at least $1 - \mathsf{SD}[A, B]$. We sometimes write $A \approx_\varepsilon B$ to say that A and B are at distance at most ε.

Recall the definition of *strong randomness extractors* [33] from a set $\mathcal{M}$.

Definition 1. A randomized function $\mathsf{Ext} : \mathcal{M} \to \{0,1\}^\ell$ with randomness of length r is an (m, ℓ, ε)-strong extractor if for all m-sources W on $\mathcal{M}$, $(\mathsf{Ext}(W; I), I) \approx_\varepsilon (U_\ell, U_r)$, where $I = U_r$ is independent of W.

We think of the output of the extractor as a key generated from $w \leftarrow W$ with the help of a seed $i \leftarrow I$. The definition states that this key behaves as if it is independent of the other parts of the system with probability $1 - \varepsilon$.

Strong extractors can extract at most $\ell = m - 2\log\left(\frac{1}{\varepsilon}\right) + \mathcal{O}(1)$ bits from (arbitrary) m-sources [35]. There has been extensive research on extractors approaching this bound, typically focusing on reducing the length of the seed (see [38] for references). For our purposes, seed length is

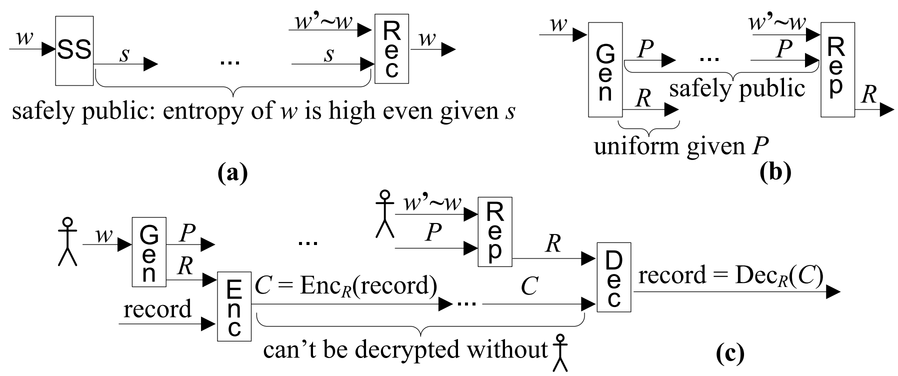


Figure 1: **(a)** secure sketch; **(b)** fuzzy extractor; **(c)** a sample application: the user encrypts a sensitive record using a key R extracted from biometric w via a fuzzy extractor; both P and the encrypted record may be sent or stored in the clear.

not too important, and universal hash functions[1] [9] are sufficient. These extract $\ell = m+2-2\log\left(\frac{1}{\varepsilon}\right)$ bits (see [41, Theorem 8.1] and references therein), and generalize to conditional min-entropy without any loss: for any E (possibly dependent on W), if $\widetilde{\mathsf{H}}_\infty(W|E) \geq m$ and $\ell = m + 2 - 2\log\left(\frac{1}{\varepsilon}\right)$, then $(\mathsf{Ext}(W;I), I, E) \approx_\varepsilon (U_\ell, I, E)$ (as shown in [18, Lemma 2.4]).Other extractors also work for conditional min-entropy, but may incur additional entropy loss.

2.2 Secure Sketches and Fuzzy Extractors

We start by presenting the basic definitions of secure sketches and fuzzy extractors from [18, Section 3],which are illustrated (together with a sample encryption application) in Figure 1. Extensions of these definitions (that handle active attacks, provide more privacy, or refine the notion of correctness) are provided in Sections 4, 5 and 6. Let $\mathcal{M}$ be a metric space with distance function dis. Informally, a secure sketch enables recovery of a string $w \in \mathcal{M}$ from any "close" string $w' \in \mathcal{M}$ without leaking too much information about w.

Definition 2. An $(m, \tilde{m}, t)$-secure sketch is a pair of efficient randomized procedures $(\mathsf{SS}, \mathsf{Rec})$ ("sketch" and "recover") such that:

1. The sketching procedure SS on input $w \in \mathcal{M}$ returns a string $s \in \{0,1\}^*$. The recovery procedure Rec takes an element $w' \in \mathcal{M}$ and $s \in \{0,1\}^*$.

2. *Correctness:* If $\mathsf{dis}(w, w') \leq t$ then $\mathsf{Rec}(w', \mathsf{SS}(w)) = w$.

3. *Security:* For any m-source over $\mathcal{M}$, the min-entropy of W given s is high: for any (W, E), if $\widetilde{\mathsf{H}}_\infty(W|E) \geq m$, then $\widetilde{\mathsf{H}}_\infty(W \mid \mathsf{SS}(W), E) \geq \tilde{m}$.

We will see several examples of secure sketches in the next section. For now, we move on to *fuzzy extractors*, which do not recover the original input, but rather enable generation of a close-to-uniform string R from w and its subsequent reproduction given any w' close to w. The reproduction is done with the help of the helper string P produced during the initial extraction; yet P need not remain secret, because R is nearly uniform even given P. Strong extractors are a special case of fuzzy extractors, corresponding to $t = 0$ and $P = I$.

[1] $\mathsf{Ext}(w;i)$ with an ℓ-bit output is universal if for each $w_1 \neq w_2$, $\mathbb{P}_i[\mathsf{Ext}(w_1;i) = \mathsf{Ext}(w_2;i)] = 2^{-\ell}$. If elements of $\mathcal{M}$ can represented as n-bit strings, such functions can be built quite easily using seeds of the length n: for instance, simply view w and x as members of $\mathrm{GF}(2^n)$, and let $\mathsf{Ext}(w;x)$ be ℓ least significant bits of wx.

Definition 3. An $(m, \ell, t, \varepsilon)$-fuzzy extractor is a pair of efficient randomized procedures $(\mathsf{Gen}, \mathsf{Rep})$ ("generate" and "reproduce") such that:

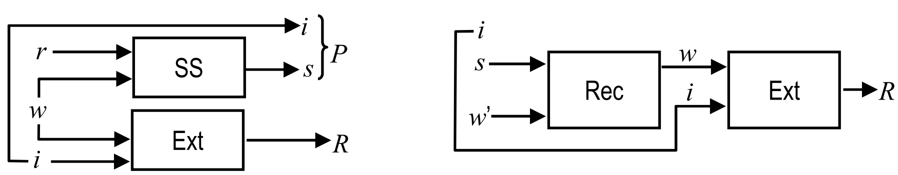
1. Gen, given $w \in \mathcal{M}$, outputs an extracted string $R \in \{0,1\}^\ell$ and a helper string $P \in \{0,1\}^*$. Rep takes an element $w' \in \mathcal{M}$ and a string $P \in \{0,1\}^*$.
2. *Correctness:* If $\mathsf{dis}(w, w') \leq t$ and $(R, P) \leftarrow \mathsf{Gen}(w)$, then $\mathsf{Rep}(w', P) = R$.
3. *Security:* For all m-sources W over $\mathcal{M}$, the string R is nearly uniform even given P, i.e., if $\widetilde{\mathsf{H}}_\infty(W \mid E) \geq m$, then $(R, P, E) \approx_\varepsilon (U_\ell, P, E)$.

Entropy loss of a secure sketch (resp. fuzzy extractor) is $m - \tilde{m}$ (resp. $m - \ell$).

We reiterate that the nearly-uniform random bits output by a fuzzy extractor can be used in a variety of cryptographic contexts that require uniform random bits (e.g., for secret keys). The slight nonuniformity of the bits may decrease security, but by no more than their distance ε from uniform. By choosing ε sufficiently small (e.g., 2^{-100}) one can make the reduction in security irrelevant. If more than ℓ random bits are needed, then pseudorandom bits can be obtained by inputting R to a pseudorandom generator [3].

2.3 Secure Sketches Imply Fuzzy Extractors

Given a secure sketch, one can always construct a fuzzy extractor which generates a key of length almost $\tilde{m}$ by composing the sketch with a good (standard) strong extractor. Because universal hash functions extract well even from conditional min-entropy, they incur the least entropy loss, and we state the result for them. Other extractors can also be used, but may lose more entropy [18, Lemma 2.3].



Lemma 1 ([18, Lemma 4.1])**.** *Suppose we compose an $(m, \tilde{m}, t)$-secure sketch $(\mathsf{SS}, \mathsf{Rec})$ for a space $\mathcal{M}$ and a universal hash function $\mathsf{Ext} : \mathcal{M} \to \{0,1\}^\ell$ as follows: in Gen, choose a random i, let $P = (\mathsf{SS}(w), i)$ and $R = \mathsf{Ext}(w; i)$; let $\mathsf{Rep}(w', (s, i)) = \mathsf{Ext}(\mathsf{Rec}(w', s), i)$. The result is an $(m, \ell, t, \varepsilon)$-fuzzy extractor with $\ell = \tilde{m} + 2 - 2\log\left(\frac{1}{\varepsilon}\right)$.*

The *random oracle model* is a heuristic for analyzing protocols with cryptographic hash functions. The idea is to mentally replace the hash function by a truly random oracle that can be queried by all parties, but only a bounded number of times. Although this model cannot be instantiated by any real hash function in general [7], it is widely used in situations where the use of real hash functions is difficult to analyze. In the random oracle model, $\mathsf{Ext}(\cdot, i)$ can be replaced with a random oracle $H(\cdot)$ in Lemma 1, and i is not needed. The extracted string can be arbitrarily long, and will appear ε-close to uniform to any observer making at most $2^{\tilde{m} - \log 1/\varepsilon}$ queries to H. This yields better parameters than Lemma 1, at the price of resorting to a heuristic analysis.

3 Basic Constructions

Secure sketches are related to the problem of information reconciliation, in which two parties holding close values $w \approx w'$ want to reconcile the differences in a communication-efficient manner. A one-message information reconciliation scheme directly yields a secure sketch (with the message being the sketch), which has low entropy loss simply because the message is short. Thus, Constructions 2 and 5 below are very similar to information-reconciliation constructions of [47, Section VI.C] (citing [39]) and [1]. Other information-reconciliation protocols [32] also yield secure sketches. We will occasionally use this communication-based perspective in our analysis.

Because of their error-tolerance, constructions of secure sketches are based on error-correcting codes, which we briefly review. Let $\mathcal{F}$ be a finite alphabet of size F. For two equal-length strings w and w', let the Hamming distance $\mathsf{dis}_{\mathsf{Ham}}(w, w')$ be the number of locations in which these strings differ (these are called "Hamming errors"). A subset $C \subset \mathcal{F}^n$ is an $[n, k, d]_{\mathcal{F}}$ code if it contains F^k elements such that $\mathsf{dis}_{\mathsf{Ham}}(v, w) \geq d$ for any $v, w \in C$. A *unique decoding* algorithm, given $w \in \mathcal{F}^n$, finds an element $c \in C$ such that $\mathsf{dis}_{\mathsf{Ham}}(c, w) \leq (d-1)/2$ if one exists (if it exists, it is unique by the triangle inequality).

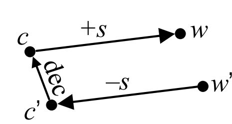
If $\mathcal{F}$ is a field and C is, additionally, a linear subspace of the vector space $\mathcal{F}^n$, then C is known as a *linear* code. In that case, fixing an $(n-k) \times n$ matrix syn over $\mathcal{F}$ whose kernel is C, a *syndrome* of a vector $w \in \mathcal{F}^n$ is defined as $\mathsf{syn}(w)$. If the Hamming weight of (i.e., number of nonzero symbols in) w is at most $(d-1)/2$, then w is unique given $\mathsf{syn}(w)$, and can be recovered by the unique decoding algorithm (e.g., by applying it to an arbitrary preimage v of $\mathsf{syn}(w)$ to get $c \in C$ and computing $w = v - c$).

3.1 Hamming Distance

To obtain a secure sketch for correcting Hamming errors over $\mathcal{F}^n$, we start with a $[n, k, 2t+1]_{\mathcal{F}}$ error-correcting code C. The idea is to use C to correct errors in w even though w may not be in C, by shifting the code so that a codeword matches up with w, and storing the shift as the sketch. View $\mathcal{F}$ as an additive group (this is automatic if $\mathcal{F}$ is a field; otherwise, use $\mathbb{Z}_F$).

Construction 1 (Code-Offset Construction)**.** On input w, select a uniformly random codeword $c \in C$, and set $\mathsf{SS}(w)$ to be the shift needed to get from c to w: $\mathsf{SS}(w) = w - c$. To compute $\mathsf{Rec}(w', s)$, subtract the shift s from w' to get $c' = w' - s$; decode c' to get c (note that because $\mathsf{dis}_{\mathsf{Ham}}(w', w) \leq t$, so is $\mathsf{dis}_{\mathsf{Ham}}(c', c)$); and compute w by shifting back to get $w = c + s$.

When the code C is linear, the information in s is essentially the syndrome of w. Then the scheme can improved to have a shorter sketch, as follows.



Construction 2 (Syndrome Construction)**.** The sketch algorithm $\mathsf{SS}(w)$ computes $s = \mathsf{syn}(w)$. To compute $\mathsf{Rec}(w', s)$, find the unique vector $e \in \mathcal{F}^n$ of Hamming weight $\leq t$ such that $\mathsf{syn}(e) = \mathsf{syn}(w') - s$, and output $w = w' - e$.

Both of these constructions have appeared in other contexts, some closely related to ours, e.g., [2, 13, 26]. They were analyzed as secure sketches in [18] (where it is also pointed out [18, Section 4.2] that the first construction generalizes to other, so called "transitive" metric spaces).

Theorem 1 ([18, Theorems 5.1, 5.2])**.** *For any m, given an $[n, k, 2t+1]_{\mathcal{F}}$ error-correcting code, Construction 1 (and Construction 2 if the code is linear) is an $(m, m-(n-k)\log F, t)$ secure sketch for the Hamming distance over $\mathcal{F}^n$. Combined with Lemma 1, these constructions give, for any ε, an $(m, m-(n-k)\log F + 2 - 2\log\left(\frac{1}{\varepsilon}\right), t, \varepsilon)$ fuzzy extractor for the same metric.*

For the syndrome construction, the analysis of entropy loss follows easily from the fact that the length of the sketch is $(n-k)\log F$ bits. A slightly more involved proof is needed for the code-offset construction.

The tradeoff between the error tolerance and the entropy loss depends on the choice of error-correcting code. For large alphabets ($\mathcal{F}$ is a field of size $\geq n$), one can use Reed-Solomon codes to get the optimal entropy loss of $2t \log F$. No secure sketch construction can have better tradeoff between error tolerance and entropy loss than Construction 1, as searching for better secure sketches for the Hamming distance is equivalent to searching for better error-correcting codes.[2] Better secure sketches, however, can be achieved if one is willing to slightly weaken the guarantee of correctness (Section 4).

While the syndrome construction may seem like only a mild improvement over the code-offset construction, the difference is crucial for the set difference metric considered in the next section.

3.2 Set Difference

We now turn to the case when inputs are relatively small subsets of a huge universe $\mathcal{U}$; let $n = |\mathcal{U}|$. This corresponds to representing an object by a list of its features. Examples include "minutiae" (ridge meetings and endings) in a fingerprint, short strings that occur in a long document, and lists of favorite movies. We stress that $\mathcal{U}$ may be very large (e.g., all 100-character strings); we consider work polynomial in $\log n$ to be feasible, but anything polynomial in n to be too expensive (for a discussion of small $\mathcal{U}$, see [18, Section 6.1]). The distance between two sets $w, w' \subseteq \mathcal{U}$ is the size of their symmetric difference: $\mathsf{dis}_{\mathsf{Set}}(w, w') \stackrel{\text{def}}{=} |w \triangle w'|$.

Here we describe the "PinSketch" construction of [18, Section 6.3].Other constructions [25, 32] have also been proposed, and are analyzed as secure sketches (with improvements to the first one) in [18, Sections 6.2, 9]. However, the PinSketch construction is the most efficient and flexible of them; it is also linear over GF(2), which will be useful when we turn to robust extractors in Section 6. A set w can be represented by its *characteristic vector* $x_w \in \{0,1\}^{\mathcal{U}}$, with 1 at positions $a \in \mathcal{U}$ if $a \in w$, and 0 otherwise. The symmetric difference of two sets is exactly the Hamming distance of their characteristic vectors. Thus, we may try applying solutions designed for the Hamming distance. In our setting, where $\mathcal{U}$ is large, this representation is too large to be written down explicitly. Nonetheless, this view of sets leads to a good construction of secure sketches.

Recall that Construction 2 for the Hamming metric results in secure sketches that are $n-k$ characters long. Thus, it is possible that the sketches will be of reasonable length even for very large n, as long as we choose a linear code for the binary alphabet with k close to n. Indeed, binary BCH codes (see, e.g. [44]) fit the bill: they are a family of $[n, n - t\alpha, 2t+1]_2$ linear codes for any t, α and $n = 2^\alpha - 1$. (These codes are optimal for $t \ll n$ by the Hamming bound, which implies that $k \leq n - \log \binom{n}{t}$ [44].) Using these codes in Construction 2 results in sketches that correct up to t set-difference errors and consist of t values of length α bits each. It is impossible to do much better, since the reconstruction algorithm may get w' that is simply missing t elements of w and would have to find them from the information in the sketch s. The entropy loss of this secure sketch will be $t\lceil \log(n+1) \rceil$.

The only problem is that the scheme appears to require computation time $\Omega(n)$, since we must compute the linear function $s = \mathsf{syn}(x_w)$ and, later, run a decoding algorithm on $\mathsf{syn}(x_{w'}) \oplus s$ to find error vector e. For BCH codes, this difficulty can be overcome.

The resulting construction works for w and w' of any size, as long their symmetric difference has at most t elements. To use it, one needs to view the universe $\mathcal{U}$ as a set of all strings of length α, except the string 0^α (this may mean enlarging the actual universe—for instance going from movie

[2] This follows from [37], which applies to secure sketches that work for the uniform distribution. We do not know of a result that rules out constructions of secure sketches which work better for lower entropy levels, though by Lemma C.1 of [18] they would not work as well for the uniform case.

titles to all nonzero strings of a certain length, whether or not they are titles of actual movies). $\mathcal{U}$ is then identified with the nonzero elements of the field $\mathrm{GF}(2^\alpha)$.

Construction 3 (PinSketch)**.**
To compute $\mathsf{SS}(w) = \mathsf{syn}(x_w)$:

1. Let $s_i = \sum_{a \in w} a^i$ (computations in $\mathrm{GF}(2^\alpha)$).
2. Output $\mathsf{SS}(w) = (s_1, s_3, s_5, ..., s_{2t-1})$.

To recover $\mathsf{Rec}(w', (s_1, s_3, \ldots, s_{2t-1}))$:

1. Compute $(s'_1, s'_3, \ldots, s'_{2t-1}) = \mathsf{SS}(w') = \mathsf{syn}(x_{w'})$.
2. Let $\sigma_i = s'_i - s_i$ (in $\mathrm{GF}(2^\alpha)$, so "$-$" is the same as "$+$").
3. Use the sublinear BCH decoding algorithm of [18, Appendix E] to find a set v of size at most t such that $\mathsf{syn}(x_v) = (\sigma_1, \sigma_3, \ldots, \sigma_{2t-1})$.
4. Output $w = w' \triangle v$.

An implementation of this construction, including the reworked BCH decoding algorithm, is available [24].

Theorem 2 ([18, Theorem 6.3])**.** *For all m, t and α, Construction 3 is an $(m, m - \alpha t, t)$ secure sketch for set difference over $\mathrm{GF}(2^\alpha)^*$. Combined with Lemma 1, this construction gives, for any ε, an $(m, m - t\alpha + 2 - 2\log\left(\frac{1}{\varepsilon}\right), t, \varepsilon)$ fuzzy extractor for the same metric.*

3.3 Edit Distance

In this section we are interested in the space of n-character strings over some alphabet $\mathcal{F}$, with distance between two strings $\mathsf{dis}_{\mathsf{Edit}}(w, w')$ defined as the smallest number of character insertions and deletions needed to get from one string to the other. It comes up, for example, when the password is entered as a string, due to typing errors or mistakes made in handwriting recognition.

It is difficult to work with this space directly. Instead, the constructions below proceed by constructing *embeddings* of edit distance into the set difference or Hamming metric which approximately preserve distance. That is, given a string w, they compute a function $\psi(w)$ with output in a simpler space, and apply one of the previous constructions to $\psi(w)$.

First, we consider the low-distortion embedding of Ostrovsky and Rabani [34]. It is an injective, polynomial-time computable mapping $\psi_{\mathsf{OR}} : \{0,1\}^n \to \{0,1\}^d$, which "converts" edit to Hamming distance. Namely, if $\mathsf{dis}_{\mathsf{Edit}}(w, w') \le t$, then $\mathsf{dis}_{\mathsf{Ham}}(\psi_{\mathsf{OR}}(w), \psi_{\mathsf{OR}}(w')) \le tD_{\mathsf{OR}}$, where $D_{\mathsf{OR}} = 2^{\mathcal{O}(\sqrt{\log n \log\log n})}$.

We now apply the syndrome construction above, using BCH codes, to get a fuzzy extractor for the edit distance over $\{0,1\}^n$. (A secure sketch can also be constructed, but to have an efficient Rec, we would need an efficient algorithm for ψ_{OR}^{-1}, whose existence is currently unknown.)

Construction 4. Let Gen' and Rep' be from Construction 1 or 2, instantiated with a $[d', k, 2tD_{\mathsf{OR}} + 1]_2$ error-correcting code. Let $\mathsf{Gen}(w) = \mathsf{Gen}'(\psi_{\mathsf{OR}}(w))$, and $\mathsf{Rep}(w', P) = \mathsf{Rep}'(\psi_{\mathsf{OR}}(w'), P))$.

Theorem 3 ([18, Proposition 7.2])**.** *Construction 4 is an $(m, \ell, t, \varepsilon)$-fuzzy extractor for the edit metric over $\{0,1\}^n$ for any n, t, m, ε and $\ell = m - t2^{\mathcal{O}(\sqrt{\log n \log\log n})} - 2\log\left(\frac{1}{\varepsilon}\right)$.*

This construction works well when the fraction of errors is small, particularly if number of errors $t = n^{1-\Omega(1)}$, because the entropy loss is $tn^{o(1)}$. *To extend this construction to a general alphabet $\mathcal{F}$, simply represent each character in $\mathcal{F}$ in binary as a string of $\lceil \log F \rceil$ bits before applying ψ_{OR}.*

An observation that allows one to handle larger error rates (and also create a much simpler construction) is that the stringent conditions traditionally placed on embeddings are not necessary to build fuzzy extractors. Informally, all that is needed is a mapping ψ that (1) does not have too many points $w_1 \neq w_2$ for which $\psi(w_1) = \psi(w_2)$ (thus causing entropy loss), and (2) for which $\mathsf{dis}_{\mathsf{Edit}}(w_1, w_2) \leq t$ implies $\mathsf{dis}(\psi(w_1), \psi(w_2)) \leq t'$ for some t' and some notion of distance that already-known fuzzy extractors can handle. Moreover, if ψ is invertible given a relatively small amount of extra information, then it can be used to build secure sketches, which will simply store the extra information as part of the sketch. The precise requirements for such ψ are spelled out in [18, Section 4.3], where they are called *biometric embeddings*.

We construct such an embedding ψ_{SH} as follows. A *c-shingle* is a length-c consecutive substring of a given string w. A *c-shingling* [6] of a string w of length n is the set (ignoring order or repetition) of all $(n - c + 1)$ c-shingles of w. (For instance, a 3-shingling of “abcdecdegh” is {abc, bcd, cde, dec, ecd, deg, egh}). Our $\psi_{\mathsf{SH}}(w)$ is simply the c-shingling of the string w for some parameter c; its output is a nonempty subset of $\mathcal{F}^c$ of size at most $n - c + 1$. A single edit error adds at most $2c - 1$ to the set difference.

In order to invert the map ψ_{SH}, we need to add information on how to reconstruct the string w from its shingles. This information, denoted by $g_c(w)$, can be computed as follows: sort $\psi_{\mathsf{SH}}(w)$ alphabetically, and let $g_c(w)$ be the list of $\lceil n/c \rceil$ indices into the set $g_c(w)$ that tell which shingle gives the first c characters, the next c characters, and so on. This leads to the following construction of secure sketches (note that there are no limitations on the lengths of w and w' in the constructions, although the resulting entropy loss will depend on $n = |w|$; in particular it may be that $|w| \neq |w'|$).

Construction 5. Fix a parameter c. Let SS' and Rec' be from Construction 3. Let $\mathsf{SS}(w) = (s', g)$, where $s' = \mathsf{SS}'(\psi_{\mathsf{SH}}(w))$ and $g = g_c(w)$. Let $\mathsf{Rec}(w', (s', g))$ be computed as the reconstruction of w from the alphabetically-ordered shingles in $\mathsf{Rec}(\psi_{\mathsf{SH}}(w'), s')$ using the indices in g.

Construction 6. One can build a fuzzy extractor from Construction 5 using Lemma 1. However, it is simpler to use the fuzzy extractor for set difference (given by Construction 3 and Lemma 1), applying Gen to $\psi_{\mathsf{SH}}(w)$ and Rep to $\psi_{\mathsf{SH}}(w')$. This saves the computation of $g_c(w)$, which is not needed, because fuzzy extractors, unlike secure sketches, do not require reconstruction of w.

Theorem 4. *For any n, m, c and ε, Construction 5 is an $(m, \tilde{m}, t)$-secure sketch, where $\tilde{m} = m - \lceil \frac{n}{c} \rceil \log_2(n-c+1) - (2c-1)t\lceil \log(F^c+1) \rceil$, and Construction 6 is an $(m, \tilde{m}+2-2\log\left(\frac{1}{\varepsilon}\right), t, \varepsilon)$-fuzzy extractor.*

The optimal choice of c results in entropy loss $m - \tilde{m} \approx 2.38(t \log F)^{1/3}(n \log n)^{2/3}$ (according to analysis in [18, Section 7.2]). If the original string has a linear amount of entropy $\theta(n \log F)$, then this construction can tolerate $t = \Omega(n \log^2 F / \log^2 n)$ insertions and deletions while extracting $\theta(n \log F) - 2 \log\left(\frac{1}{\varepsilon}\right)$ bits. The number of bits extracted is linear; if the string length n is polynomial in the alphabet size F, then the number of errors tolerated is linear also.

3.4 Other Distance Measures

There have been several works constructing secure sketches (or variants thereof) for other distance measures. We list here a few examples: Chang and Li [10] constructed the first non-trivial sketches for “point-set difference,” an abstraction of several distance measures commonly used for matching fingerprints. Linnartz and Tuyls [29] (see also [45, 42] and Chapter 16 of [43]) considered key extraction primitives for continuous metrics, assuming the biometric is distributed according to a known multidimensional Gaussian distribution. Li, Sutcu and Memon [28] explored the question of building secure sketches for discrete distributions in continuous spaces. They showed that a large class of quantizers are equivalent from the entropy point of view, simplifying the design of sketches for such spaces.

4 Improving Error-Tolerance via Relaxed Notions of Correctness

So far, we required *perfect correctness*, i.e., the primitives worked for *every* w' within distance t of w, with no probability of error. In [18, Section 8] it is shown how the number of tolerated errors t can be significantly increased by insisting only that errors be corrected with very high probability. In other words, we allow that for an unlikely choice of w', the correction will fail.

As an illustration, consider binary Hamming errors. The Plotkin bound from coding theory implies that no secure sketch with residual entropy $\tilde{m} \geq \log n$ can correct $n/4$ errors with probability 1. However, under a relaxed notion of correctness (made precise below), one can construct secure sketches that tolerate arbitrarily close to $n/2$ errors, i.e., correct $n(\frac{1}{2} - \gamma)$ errors for any constant $\gamma > 0$, and still have residual min-entropy $\tilde{m} = \Omega(n)$ (provided the original min-entropy $m \geq (1 - \gamma')n$, for some $\gamma' > 0$ dependent on γ).

We discuss relaxed correctness arising from considering the following error models: random errors, input-dependent errors and computationally-bounded errors. For simplicity, we limit our discussion to binary Hamming errors.

Random Errors. Assume there is a *known* distribution on the errors which occur in the data. The most common distribution is the binary symmetric channel BSC_p: each bit of the input is flipped with probability p and left untouched with probability $1 - p$. In that case, error-correcting codes, and by extension the code-offset construction, can correct an error rate up to Shannon's bound on noisy channel coding, and no more (see, e.g., [12] for a textbook coverage of Shannon's coding bounds). If $p \in (0, 1/2)$, there are deterministic sketches with entropy loss $n\mathsf{h}(p) - o(n)$ where h is the binary entropy function $\mathsf{h}(p) = -p \log p - (1-p) \log(1-p)$. In particular, if $m \geq n(\mathsf{h}(\frac{1}{2} - \gamma) + \varepsilon)$, which is strictly less than n for a small enough ε, one can tolerate $(\frac{1}{2} - \gamma)$-fraction of errors, and still have residual min-entropy $m' \geq \varepsilon n$.

The assumption of a known noise process is, for our taste, far too strong: there is no reason to believe we understand the exact distribution on errors which occur in complex data such as biometrics.[3] However, it provides a useful baseline by which to measure results for other models.

Input-dependent Errors. Here the errors are adversarial, subject only to the conditions that (a) the error $\mathsf{dis}(w, w')$ is bounded to a maximum magnitude of t, and (b) the corrupted word *depends only on the input* w, and not on the secure sketch $\mathsf{SS}(w)$. (In particular, this means that the sketch must be probabilistic; thus, the assumption states that the errors are independent of the randomness used to compute $\mathsf{SS}(w)$.) Here we require that for any pair w, w' at distance at most t, we have $\mathsf{Rec}(w', \mathsf{SS}(w)) = w$ with high probability over the random choices made by the randomized sketching algorithm SS.

This model encompasses any complex noise process which has been observed to never introduce more than t errors. Unlike the assumption of a particular distribution on the noise, the bound on magnitude can be checked experimentally. For binary Hamming errors, perhaps surprisingly, in

[3] Since the assumption here only plays a role in correctness, it is still more reasonable than assuming we know exact distributions on the data in proofs of *secrecy*. However, in both cases, we would like to enlarge the class of distributions for which we can provably satisfy the definition of security.

this model we can tolerate exactly the same error rate t/n as in the model of random errors. That is, we can tolerate an error rate up to Shannon's coding bound and no more. Thus, *removing the assumption of a known distribution comes at no cost*.

The basic idea, originally due to Lipton [30], is to prepend a random permutation $\pi : [n] \to [n]$ to the sketch, which acts on the bit positions of the word $w = (w_1, ..., w_n) \in \{0, 1\}^n$. That is, define $\mathsf{SS}(w; \pi) = (\pi,\ \mathsf{syn}_C(w_{\pi(1)}, w_{\pi(2)}, ..., w_{\pi(n)}))$. If the difference between w and w' does not depend on the sketch, then the errors that the code C is asked to correct will in fact be randomly distributed (subject only to having a fixed number of bits flipped). Thus one can use codes for random errors, and reduce the entropy loss to near the Shannon limit. One can in fact further reduce the total size of the sketch by choosing π pseudo-randomly [30, 40].

Computationally-bounded Errors. Here the errors are adversarial and may depend on both w and the publicly stored information $\mathsf{SS}(w)$. However, we assume that the errors are introduced by a process of bounded computational power. That is, there is a probabilistic circuit of polynomial size (in the length n) which computes w' from w. Results proved in this model are extremely powerful: unless the process introducing errors – natural or artificial – is, say, capable of forging digital signatures, then the sketch will successfully recover from the errors. It is not clear, however, whether this model allows correcting binary Hamming errors up to the Shannon bound, as in the two models above. As we explain below, the question is related to some open questions regarding the constructions of efficiently *list-decodable* codes. These codes enlarge the radius of tolerated errors in return for generating a list of not one, but polynomially-many codewords. However, when the error rate t/n is either very high or very low, then the appropriate list-decodable codes exist and we can indeed match the Shannon bound.

The main technique is to use the syndrome of a list-decodable code in the Code-Offset construction, as opposed to the usual, uniquely-decodable code. Appending a short hash or a digital signature of the original input w to the syndrome allows the recovery algorithm to disambiguate the (polynomially long) list and decide which of the possibilities leads to the correct value of w. This allows tolerating arbitrary errors as long as the process introducing the errors is unable to "break" (that is, find a collision for) the hash function or the digital signature. This idea of sieving is fundamental to almost all applications of list-decoding. See [40, Table 1] for a summary of these ideas as they apply to information reconciliation and secure sketches.

5 Strong Privacy Guarantees

Although secure sketches leave entropy in w, and fuzzy extractors even extract uniform randomness from it, they may leak (through the string s or P) sensitive information about w: an adversary may learn a function $f(w)$ such as the user's eye color, ethnicity, or risk of diabetes. Moreover, when the very process that generates w is secret (this does not happen when w is just a user's biometric, but happens in example applications considered at the end of this section), s or P may leak information about that process itself, with undesirable consequences for the future. We therefore ask if it is possible to design fuzzy extractors which leak "no information" about w.

To get some intuition, let us first consider the noiseless case (i.e., when $t = 0$). Strong extractors directly yield fuzzy extractors whose helper string $P = i$ (the seed to a strong extractor) is independent of w. However, for $t > 0$, such a construction is impossible: the mutual information between w and P has to be non-trivial (and to grow roughly proportionally to t) [19]. In other words, P must leak *some* information about w in order to help correct errors in w. In contrast, it turns out that one can correct errors in w without leaking anything *"useful"* about w. We need the following definition:

Definition 4. ([20]) A probabilistic map $Y()$ *hides all functions of* W with leakage ε if for every adversary $\mathcal{A}$, there exists an adversary $\mathcal{A}_*$ such that for all functions $f : \{0,1\}^* \to \{0,1\}^*$,

$$\left|\Pr[\mathcal{A}(Y(W)) = f(W)] - \Pr[\mathcal{A}_*() = f(W)]\right| \le \varepsilon.$$

The map $Y()$ is called (m, ε)-*entropically secure* if $Y()$ hides all functions of W, for all m-sources W.

In our context, a natural choice for the variable $Y(W)$ should be the helper string $P = P(W)$. In fact, we can do better and set $Y = \mathsf{Gen}(W) = (R, P)$. As long as W has sufficient min-entropy, it is nearly as hard to predict $f(W)$ given *both* R *and* P, as it is without (R, P), regardless of the adversary's computing power. This strengthening is important since the "outside" application which uses R as its secret key may leak R (think of a one-time pad, for example). We ensure the *information-theoretic* privacy of w even if R is eventually leaked or used as a key in a computationally secure scheme:

Definition 5. A $(m, \ell, t, \varepsilon)$ fuzzy extractor for $\mathcal{M}$ is called (m, ε')-private if the pair (R, P) is (m, ε')-entropically secure.

Entropic security should not be confused with the much stronger notion of statistical "Shannon" secrecy, which requires that (R, P) and W are (almost) statistically independent*(as already mentioned, such a strong notion is impossible to achieve, even when restricted to* P *alone)*. Entropic security only states that (R, P) does not help in predicting $f(W)$, for any function f *specified in advance*. For example, after seeing any particular realization (r, p) of (R, P), one might learn many functions $f_{(r,p)}$ of W. Nonetheless, entropic security is quite strong: no particular function f is likely to be one of them for random (r, p). Shannon security is impossible to achieve in our context, and entropic security provides an alternative which is sufficient in several applications.

Uniform Fuzzy Extractors. A seemingly weaker definition than entropic security is that of (m, ε)-indistinguishability [20]: a (probabilistic) map $Y()$ is (m, ε)-*indistinguishable* if for all pairs of m-sources W_1, W_2, the distributions $Y(W_1)$ and $Y(W_2)$ are ε-close. *Indistinguishable maps hide all information about the* distribution W*: all possible distribution* $Y(W)$ *are* ε*-close to the single distribution* $Y(U_n)$. For example, extractors for m-sources are trivially $(m, 2\varepsilon)$-indistinguishable. Entropic security is essentially equivalent to indistinguishability: hiding all functions of W is nearly the same as hiding the distribution of W.

Theorem 5 ([20])**.** *If* $Y()$ *is* (m, ε)*-entropically secure, then it is* $(m-1, 4\varepsilon)$*-indistinguishable. Conversely, if* $Y()$ *is* (m, ε)*-indistinguishable, then it is* $(m+2, 8\varepsilon)$*-entropically secure. In particular, extractors for min-entropy* m *hide all functions for sources of min-entropy* $m + 2$*.*

Applying this characterization to the case of private fuzzy extractors, we see that it is enough to construct what we call a (m, ε)-*uniform* fuzzy extractor, which is defined by the constraint $(R, P) \approx_\varepsilon (U_\ell, U_{|P|})$. A uniform extractor automatically satisfies the security requirement on fuzzy extractors. By Theorem 5, it is also $(m + 2, 8\varepsilon)$-private. Thus, we focus our attention on uniform fuzzy extractors.

Extractor-Sketches. Similar to the definitions of private and uniform fuzzy extractors, we can also define private or uniform secure sketches. For our purposes, however, it will be convenient to define the following special case of uniform (and, therefore, private) secure sketches.

Definition 6. A $(m, \tilde{m}, t, \varepsilon)$-*extractor-sketch* (for $\mathcal{M}$) is given by two efficient randomized procedures $(\mathsf{Ext}, \mathsf{Rec})$ s.t. (a) Ext is a $(m, \lambda, \varepsilon)$-strong extractor (for some λ, typically equal to $m - \tilde{m}$; see below), and (b) $(\mathsf{SS}, \mathsf{Rec})$ is a $(m, \tilde{m}, t)$-secure sketch, where $\mathsf{SS}(W; I) \stackrel{\text{def}}{=} (\mathsf{Ext}(W; I), I)$.

As the definition states, extractor-sketches are simultaneously strong extractors and secure sketches. On the one hand, the output of the sketch looks uniform: $(\mathsf{Ext}(W; I), I) \approx_\varepsilon U_{|\mathsf{SS}(W)|}$. In particular, an extractor-sketch of W hides the distribution and all the functions of W. On the other hand, w can be recovered from $\mathsf{Ext}(w; i)$, the seed i and any w' close to w. Unlike the standard rationale for extractors, however, the objective of such extractor-sketches is to *minimize* their output length λ, since this length corresponds to the length of the secure sketch, which directly bounds the entropy loss $m - \tilde{m} \leq \lambda$ of the sketch (in fact, in most constructions $\lambda = m - \tilde{m}$). In other words, the purpose of extractor-sketches is the recovery of w using the minimal amount of information and not randomness extraction. The latter property only serves to argue privacy via Theorem 5.

We now make a few observations. First, every $(m, \tilde{m}, t, \varepsilon)$-extractor-sketch is $(m+2, 8\varepsilon)$-private by Theorem 5, meaning that no function about a source of min-entropy $m + 2$ is leaked. Second, unlike ordinary secure sketches, extractor-sketches must be probabilistic (otherwise they cannot be entropically secure). Third, fuzzy extractors constructed from secure sketches via Lemma 1 will be $(m, \varepsilon + \varepsilon')$-uniform if an $(m, \tilde{m}, t, \varepsilon')$-extractor-sketch is composed with an ordinary universal hash function to extract $\ell = \tilde{m} + 2 - 2\log\left(\frac{1}{\varepsilon}\right)$ bits. Thus, in the sequel we will only need to construct extractor-sketches. We limit our discussion to the Hamming metric.

Constructing Extractor-Sketches. Dodis and Smith [19] constructed extractor-sketches for the Hamming space $\mathcal{F}^n$, where $|\mathcal{F}| = F$. The actual construction of extractor-sketches in [19] uses a special family $\{C_i\}_i$ of $[n, k, d = 2t + 1]$-codes (for "appropriate" k) over $\mathcal{F}^n$, and sets $\mathsf{SS}(w; i) = (i, \mathsf{syn}_{C_i}(w))$. Thus, it replaces the fixed $[n, k, d = 2t + 1]$-code in the Syndrome construction from Section 3.1 by a carefully chosen family of codes $\{C_i\}_i$. The challenge was to obtain the largest possible dimension k such that, for a random code C_I and for any m-source W, the pair $(I, \mathsf{syn}_{C_I}(W))$ is ε-close to uniform.

When the alphabet size $F > n$, the family of codes used by [19] is straightforward. We start from a fixed code C equal to the $[n, n-2t, 2t+1]$-Reed-Solomon code. Given an index $i = (a_1, \ldots a_n)$ consisting of *non-zero* elements of $\mathcal{F}$, we define $C_i = \{(a_1 \cdot c_1, \ldots, a_n \cdot c_n) \in \mathcal{F}^n \mid (c_1, \ldots, c_n) \in C\}$. (Restricting a_j's to be non-zero ensures that each C_i still has minimal distance $2t+1$.) The resulting family[4] is $\left\{C_{(a_1,\ldots,a_n)} \mid a_1 \neq 0, \ldots, a_n \neq 0\right\}$. For general alphabets, including the binary alphabet $\mathcal{F} = \{0, 1\}$, the construction is somewhat more complicated, and we do not include it here.

Theorem 6 ([19]). *There exists families of efficiently encodable and decodable codes $\{C_i\}_i$ such that the above "randomized" Syndrome construction yields $(m, \tilde{m}, t, \varepsilon)$-extractor-sketches for $\mathcal{F}^n$ with the following parameters:*

(Binary Alphabet; $F = 2$) *For any min-entropy $m = \Omega(n)$, one achieves $\tilde{m}$, t and $\log\left(\frac{1}{\varepsilon}\right)$ all $\Omega(n)$, and $|\mathsf{SS}(w)| < 2n$.*

(Large Alphabet; $F > n$) *For $m > 2t \log F$, we have $\tilde{m} = m - 2t \log F$ and $\varepsilon = \mathcal{O}(2^{-\tilde{m}/2})$. Both of these parameters are optimal. The total length of the sketch is $|\mathsf{SS}(w)| = (n + 2t) \log F$.*

Thus, extractor-sketches for high alphabets can achieve the same entropy loss $\lambda = 2t \log F$ as "non-private" sketches, at the price of increasing the length of the sketch from $2t \log F$ to $(n + 2t) \log F$. Moreover, they achieve the same error $\varepsilon = \mathcal{O}(2^{-\tilde{m}/2})$ as the best strong extractors for a given output length $\lambda = m - \tilde{m}$. Thus, their only disadvantage comes from the fact that the length of the sketch is greater than the length of the message. For binary alphabets, the construction of [19] loses significant constant factors as compared to the best non-private sketches, although it is still asymptotically optimal. In particular, using Theorem 6 we obtain (m, ε)-private

[4] The code families use in the Dodis-Smith construction are sometimes called "generalized Reed-Solomon" codes.

$(m, \ell, t, \varepsilon)$-fuzzy extractors for the Hamming space $\{0,1\}^n$ where $\ell, \log\left(\frac{1}{\varepsilon}\right)$ and t are $\Omega(n)$ as long as $m = \Omega(n)$.

Applications. As we already stated, extractor-sketches immediately yield corresponding uniform (and, therefore, private) secure sketches and fuzzy extractors.

As another application of extractor-sketches, [19] showed how to use them to construct so called (m, t, ε)-*fuzzy perfectly one-way* hash functions for the binary Hamming metric. Such functions come with a fixed verification procedure Ver and allow one to publish a probabilistic hash y of the secret w such that: (a) $\mathsf{Ver}(w', y)$ will accept any w' within distance t from w; (b) it is computationally infeasible to find y and secrets w and w' of distance greater than $2t$ from each other, such that $\mathsf{Ver}(w, y)$, $\mathsf{Ver}(w', y)$ both accept; and (c) the map $W \mapsto Y$ is (m, ε)-entropically secure. Thus, publishing y allows anybody to test (without either false positives or false negatives!) whether or not some input w' is *close* to the "correct" secret input w, and yet without leaking any particular function of w. The construction of [19] achieved, for any entropy level $m = \Omega(n)$, an (m, t, ε)-fuzzy perfect one-way family where both t and $\log\left(\frac{1}{\varepsilon}\right)$ are $\Omega(n)$.

We next present two applications of private fuzzy extractors to situations when the very process that generates high-entropy w must remain secret. In the approach proposed by [8] for mitigating dictionary attacks against password-protected local storage [8], one stores on disk many different puzzles solvable only by humans (so called "captchas" [46]). A short password specified by the user selects a few puzzles to be manually solved by the user; their solutions are $w = w_1 \dots w_n$. The string w is then used to derive a secret key R (for example used to encrypt the local hard drive). The security of the system rests on the fact that the adversary does not know which puzzles are selected by the user's password: thus, the process that generates w must remain secret. In order to tolerate a few incorrectly solved puzzles, one would like to use a fuzzy extractor to derive R from w. However, an ordinary fuzzy extractor does not suffice here, because, although it will not leak too much information about w, it may leak *which puzzles* are used to construct w, and thus allow the adversary to focus on solving them and obtain w on its own. A private fuzzy extractor, however, will work.

In a similar vein, [19] showed that entropically secure sketches can be used to simultaneously achieve error correction, key reuse and "everlasting security" in the so-called bounded storage model (BSM) [31]. This resolved the main open problem of [15]. We refer to [19] for more details and references regarding this application.

6 Robustness and Protection Against Active Attacks

So far in this survey we have implicitly focused on situations with a *passive* adversary: specifically, we assumed that the adversary could not tamper with the helper string P output by the fuzzy extractor. This is not representative of all the situations where biometrics are used: users may be connected by an insecure (i.e. unauthenticated) channel, and the recovery algorithm Rep may receive an input $\tilde{P} \neq P$. Such situations call, essentially, for *authentication* based on noisy secrets $W \approx W'$. The literature has considered a variety of settings. These fall into two broad categories.

Authenticated Key Agreement: Here, a trusted server stores the "actual" biometric data W of a user; periodically, the user obtains a fresh biometric scan W' which is close, but not identical, to W. The server and user then wish to mutually authenticate and agree on a key R over a network controlled by an adversary.

Key Encapsulation: Here, a user (on his own) uses his biometric data W to generate a random key R along with some public information P, and then stores P on an untrusted disk. The key R may then be used, say, to encrypt some file for long-term storage. At a later point in time, the user obtains a fresh biometric scan W'. After retrieving P from the disk, the user recovers R (with which he can decrypt the file). The challenge here is that the value P provided by the disk may not be the same as what the user originally stored. In this second setting the user is essentially running a key agreement protocol with *himself* at two points in time, with the (untrusted) disk acting as the network. This inherently requires a *non-interactive* (i.e., one-message) key agreement protocol, since W is no longer available at the later point in time. Any solution for the second scenario is also a solution for the first.

Robust Fuzzy Extractors One can construct protocols for both these settings using a common abstraction, a *robust* fuzzy extractor, which recovers R if and only if it is fed the original helper string P produced by Gen. In all other cases, it rejects with high probability and outputs a special symbol $\perp$. The definition is illustrated in Fig. 2. It is inspired by the definition of chosen-ciphertext security of standard key encapsulation mechanisms [21]: if all variations in P are rejected, the adversary $\mathcal{A}$ is essentially constrained to behave passively. Note that the adversary is given access to the extracted key R when attempting to forge a valid $\tilde{P} \neq P$. This guarantee is necessary, for example, in the second setting above since the adversary may attempt to forge $\tilde{P}$ after seeing how R is used to encrypt a possibly known file. See [17] for a statement of the definition and discussion of the variations considered in the literature. In the remainder of this section, we sketch two constructions of robust fuzzy extractors.

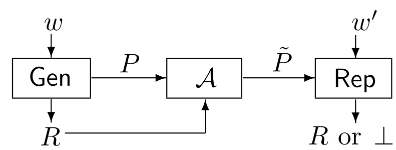


Figure 2: Robust Extractors

A Construction Using Ideal Hash Functions In the random oracle model (described in Section 2.3), there is a simple, generic construction of a robust extractor from any secure sketch. Assume we are given two hash functions $H_1, H_2 : \{0,1\}^* \to \{0,1\}^\ell$ (the pair can be simulated given a single random oracle by prepending a bit to the input).

Construction 7 ([5])**.** • $\mathsf{Gen}(w)$: Let $s = \mathsf{SS}(w)$. Output $P = (s, H_1(w,s))$ and $R = H_2(w,s)$.

- $\mathsf{Rep}(w', \tilde{P})$: Parse $\tilde{P}$ as $(\tilde{s}, \tilde{h})$ and set $\tilde{w} = \mathsf{Rec}(w', \tilde{s})$. If $\mathsf{dis}(\tilde{w}, w') \leq t$ and $\tilde{h} = H_1(\tilde{w}, \tilde{s})$ then output $H_2(\tilde{w}, \tilde{s})$, else output $\perp$.

We formulated the security definitions above for the case of computationally unbounded adversaries, and we are now discussing a protocol which is only computationally secure. Nonetheless, in the random oracle model the definitions make sense; it suffices to restrict the number of calls the adversary makes to the random oracle.

Let $\mathsf{Vol}_t^{\mathcal{M}}$ be the maximum volume of a ball of radius t in the space $\mathcal{M}$. The statement below captures the basic properties of the construction; see [5] for a more detailed statement and proof.

Proposition 1 ([5, Thm 1])**.** *Suppose that* SS *is a* $(m, \tilde{m}, t)$*-secure sketch. An adversary which makes at most* q *queries to the random oracle has forgery probability at most* $\delta = \frac{q^2}{2^\ell} + \frac{3q + 2\mathsf{Vol}_t^{\mathcal{M}}}{2^{\tilde{m}}}$*, and advantage at most* $q/2^{\tilde{m}}$ *in distinguishing* R *from random.*

Unconditionally Secure Constructions Analyses of protocols in the random oracle model are fraught with peril: they yield at best a heuristic estimate of the protocols' security (at worst, a drastic overestimate [7]). At the cost of some loss of parameters, for specific metrics one can in fact construct unconditionally-secure robust fuzzy extractors.

If we consider the random oracle construction above slightly more abstractly, we can view the first hash value (appended to P) as some kind of message authentication code (MAC) applied to the sketch s, using w as a key. We may try to emulate this in the standard model by replacing the random oracle with a regular MAC. The problem with analyzing such a scheme is that the key used by Rep to verify the MAC actually depends on the message whose authenticity is being tested. In the language of cryptanalysis, the adversary can mount a specific kind of *related key attack*. The circularity is broken in the random oracle construction above since the dependency goes through a large random table to which the adversary does not have complete access. It turns out that we can also break the circularity in the standard model, by looking carefully at the class of relations the adversary can induce on the keys and tailoring the MAC to this particular application.

Consider the case of Hamming distance in $\{0,1\}^n$. If the sketch is the syndrome with respect to a linear code (Construction 2), then the adversary controls, based on the putative sketch $\tilde{s}$, the difference $s - \tilde{s} = \mathsf{syn}(w) - \mathsf{syn}(\tilde{w})$, where $\tilde{w}$ is the candidate value for w reconstructed by Rep. If given only a few extra bits of information, namely the value $\mathsf{syn}(w')$, the adversary could in fact compute the difference in the original vectors $w - \tilde{w}$ exactly. That means that whatever value $\tilde{s}$ he provides, the adversary essentially knows the *additive* relationship between the key used to create the tag and the key used to verify it. It suffices therefore to design a MAC secure against additive related key attacks.

Fix a finite field $\mathcal{F}$. Given a message $M = (m_1, ..., m_n) \in \mathcal{F}^n$, where $n + 1 \neq 0 \mod char(\mathcal{F})$ and a uniformly random key $a, b \in \mathcal{F}$, the following is indeed secure against additively related key attacks:

$$\mathrm{MAC}_{a,b}(M) = a^{n+2} + \sum_{i=1}^{n} m_n a^n + b\,.$$

This differs from a standard information-theoretically secure constructiononly in the leading term a^{n+2}. This term causes additively related keys to induce very different polynomials, and means that the value of the tag remains unpredictable. Crucially, the security holds even when the key a, b is not uniform, but drawn from a sufficiently high-entropy distribution.

The second major problem comes from the fact that the MAC key is itself non-uniform, and there is no shared perfect randomness with which one can extract. Again, in the random oracle construction above, one circumvents this difficulty because the key space is irrelevant; each key essentially defines a completely random function. It turns out that this difficulty, too, can be circumvented for a class of information-theoretic MACs by looking carefully at the set of constraints on the key space defined by the adversary's view, and showing that the remaining set of keys is rich enough to make forgery unlikely.

The full construction of a robust extractor based on these ideas is quite delicate, and we refer the reader to [17] for details. Note however, that the construction has a fundamental limit: one cannot extract robustly if the entropy m is less than $n/2$. This limit holds for any information-theoretically secure scheme, but not for the random oracle construction described above. Finding a scheme which breaks this bound in the standard (computational) model remains an open question.

Remarks on Extensions and Other Models

1. As mentioned above, constructing robust extractors gives non-interactive protocols for both of the settings mentioned at the beginning of the section. However, it is possible to get considerably better solutions in the interactive setting. By modifying known password-authenticated key exchange protocols, one can get solutions which are in fact secure against "offline" attacks without the large entropy requirements of information-theoretically secure solutions [5].

2. The problem of authentication based on noisy secrets arises in the so-called "bounded storage model" when the long, shared string is noisy [15]. In that setting, it is reasonable to assume that the two parties Gen and Rep additionally share a short, truly random secret key. This

does not trivialize the problem, since Gen extracts a much longer key than the small shared key it starts with. Also, we require the output of Gen to leak no information about the small shared key, so that the small key can be reused. One can construct very good robust fuzzy extractors in that model, and, combined with extractor-sketches of Section 5, these yield the first stateless protocols for the bounded storage model with noise that are secure against active adversaries [17].

3. The issue of robustness is closely related to the issue of reusability, introduced by Boyen [4], and discussed in [43, Chapter 6]. Essentially, we have considered a setting where the public information P is generated once, and then possibly reused many times. One runs into much more difficult questions if many copies of P are generated *independently*. Many of those questions remain open, for example: can one construct reusable entropically-secure extractors [19]?

References

[1] S. Agarwal, V. Chauhan, and A. Trachtenberg. Bandwidth efficient string reconciliation using puzzles. *IEEE Transactions on Parallel and Distributed Systems*, 17(11):1217–1225, 2006.

[2] C. H. Bennett, G. Brassard, C. Crépeau, and M.-H. Skubiszewska. Practical quantum oblivious transfer. In J. Feigenbaum, editor, *Advances in Cryptology—CRYPTO '91*, volume 576 of *LNCS*, pages 351–366. Springer-Verlag, 1992, 11–15 Aug. 1991.

[3] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM Journal on Computing*, 13(4):850–863, Nov. 1984.

[4] X. Boyen. Reusable cryptographic fuzzy extractors. In *11th ACM Conference on Computer and Communication Security*. ACM, Oct. 25–29 2004.

[5] X. Boyen, Y. Dodis, J. Katz, R. Ostrovsky, and A. Smith. Secure remote authentication using biometric data. In R. Cramer, editor, *Advances in Cryptology—EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 147–163. Springer-Verlag, 2005.

[6] A. Broder. On the resemblence and containment of documents. In *Compression and Complexity of Sequences*, 1997.

[7] R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, pages 209–218, Dallas, Texas, 23–26 May 1998.

[8] R. Canetti, S. Halevi, and M. Steiner. Mitigating dictionary attacks on password-protected storage. In Dwork [22], pages 160–179.

[9] J. Carter and M. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18:143–154, 1979.

[10] E.-C. Chang and Q. Li. Hiding secret points amidst chaff. In S. Vaudenay, editor, *Advances in Cryptology—EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 59–72. Springer-Verlag, 2006.

[11] B. Chor and O. Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM Journal on Computing*, 17(2):230–261, 1988.

[12] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications, 1991.

[13] C. Crépeau. Efficient cryptographic protocols based on noisy channels. In W. Fumy, editor, *Advances in Cryptology—EUROCRYPT 97*, volume 1233 of *LNCS*, pages 306–317. Springer-Verlag, 1997.

[14] J. Daugman. How iris recognition works. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):21–30, 2004.

[15] Y. Z. Ding. Error correction in the bounded storage model. In Kilian [27].

[16] Y. Z. Ding, P. Gopalan, and R. J. Lipton. Error correction against computationally bounded adversaries. Manuscript. Appeared initially as [30], 2004.

[17] Y. Dodis, J. Katz, L. Reyzin, and A. Smith. Robust fuzzy extractors and authenticated key agreement from close secrets. In Dwork [22], pages 232–250.

[18] 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. arXiv:cs/0602007.

[19] Y. Dodis and A. Smith. Correcting errors without leaking partial information. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 654–663. ACM, 2005.

[20] Y. Dodis and A. Smith. Entropic security and the encryption of high entropy messages. In Kilian [27].

[21] D. Dolev, C. Dwork, and M. Naor. Nonmalleable cryptography. *SIAM*, 30:391–437, 2000.

[22] C. Dwork, editor. *Advances in Cryptology—CRYPTO 2006*, volume 4117 of *LNCS*. Springer-Verlag, 20–24 Aug. 2006.

[23] C. Dwork. Differential privacy. In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, editors, *Automata, Languages and Programming: 33rd International Colloquium (ICALP 2006), Part II*, volume 4052 of *LNCS*, pages 1–12. Springer-Verlag, 2006.

[24] K. Harmon and L. Reyzin. An implementation of syndrome encoding and decoding for binary BCH codes, secure sketches and fuzzy extractors. Available at `http://www.cs.bu.edu/~reyzin/code/fuzzy.html`.

[25] A. Juels and M. Sudan. A fuzzy vault scheme. In *IEEE International Symposium on Information Theory*, 2002.

[26] A. Juels and M. Wattenberg. A fuzzy commitment scheme. In *Sixth ACM Conference on Computer and Communication Security*, pages 28–36. ACM, Nov. 1999.

[27] J. Kilian, editor. *First Theory of Cryptography Conference — TCC 2005*, volume 3378 of *LNCS*. Springer-Verlag, 2005.

[28] Q. Li, Y. Sutcu, and N. Memon. Secure sketch for biometric templates. In *Advances in Cryptology—ASIACRYPT 2006*, volume 4284 of *LNCS*, Shanghai, China, 3–7 Dec. 2006. Springer-Verlag.

[29] J.-P. M. G. Linnartz and P. Tuyls. New shielding functions to enhance privacy and prevent misuse of biometric templates. In *AVBPA*, pages 393–402, 2003.

[30] R. J. Lipton. A new approach to information theory. In P. Enjalbert, E. W. Mayr, and K. W. Wagner, editors, *STACS*, volume 775 of *Lecture Notes in Computer Science*, pages 699–708. Springer, 1994. The full version of this paper is in preparation [16].

[31] U. Maurer. Conditionally-perfect secrecy and a provably-secure randomized cipher. *Journal of Cryptology*, 5(1):53–66, 1992.

[32] Y. Minsky, A. Trachtenberg, and R. Zippel. Set reconciliation with nearly optimal communication complexity. *IEEE Transactions on Information Theory*, 49(9):2213–2218, 2003.

[33] N. Nisan and D. Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–53, 1996.

[34] R. Ostrovsky and Y. Rabani. Low distortion embeddings for edit distance. In *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing*, pages 218–224, Baltimore, Maryland, 22–24 May 2005.

[35] J. Radhakrishnan and A. Ta-Shma. Bounds for dispersers, extractors, and depth-two superconcentrators. *SIAM Journal on Computing*, 13(1):2–24, 2000.

[36] R. Renner and S. Wolf. Simple and tight bounds for information reconciliation and privacy amplification. In B. Roy, editor, *Advances in Cryptology—ASIACRYPT 2005*, LNCS, Chennai, India, 4–8 Dec. 2005. Springer-Verlag.

[37] L. Reyzin. Entropy Loss is Maximal for Uniform Inputs. Technical Report BUCS-TR-2007-011, CS Department, Boston University, 2007. Available from `http://www.cs.bu.edu/techreports/`.

[38] R. Shaltiel. Recent developments in explicit constructions of extractors. *Bulletin of the EATCS*, 77:67–95, 2002.

[39] D. Slepian and J. K. Wolf. Noiseless coding of correlated information sources. *IEEE Transactions on Information Theory*, IT-19:471–480, July 1973.

[40] A. Smith. Scrambling aversarial errors using few random bits. In H. Gabow, editor, *SIAM-ACM Symposium on Discrete Algorithms (SODA)*, 2007.

[41] D. R. Stinson. Universal hash families and the leftover hash lemma, and applications to cryptography and computing. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 42:3–31, 2002. Available at `http://www.cacr.math.uwaterloo.ca/~dstinson/publist.html`.

[42] P. Tuyls and J. Goseling. Capacity and examples of template-protecting biometric authentication systems. In D. Maltoni and A. K. Jain, editors, *ECCV Workshop BioAW*, volume 3087 of *Lecture Notes in Computer Science*, pages 158–170. Springer, 2004.

[43] P. Tuyls, B. Skoric, and T. Kevenaar, editors. *Security with Noisy Data*. Springer-Verlag, 2007.

[44] J. van Lint. *Introduction to Coding Theory*. Springer-Verlag, 1992.

[45] E. Verbitskiy, P. Tuyls, D. Denteneer, and J.-P. Linnartz. Reliable biometric authentication with privacy protection. In *Proc. 24th Benelux Symposium on Information theory*, 2003.

[46] L. von Ahn, M. Blum, N. Hopper, and J. Langford. Captcha: Using hard ai problems for security. In E. Biham, editor, *Advances in Cryptology—EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 294–311. Springer-Verlag, 2003.

[47] A. D. Wyner. Recent results in Shannon theory. *IEEE Transactions on Information Theory*, IT-20(1):2–10, Jan. 1974.

Index