An extended abstract of this paper appears in the proceedings of the 11th ACM Conference on Computer and Communications Security—CCS 2004, Washington, DC, October 2004. This expanded version is available from the IACR Cryptology ePrint Archive at `http://eprint.iacr.org/2004/358/`.

Reusable Cryptographic Fuzzy Extractors

Xavier Boyen

Voltage Security, Palo Alto
xb@boyen.org

Abstract

We show that a number of recent definitions and constructions of fuzzy extractors are not adequate for multiple uses of the same fuzzy secret—a major shortcoming in the case of biometric applications. We propose two particularly stringent security models that specifically address the case of fuzzy secret reuse, respectively from an outsider and an insider perspective, in what we call a chosen perturbation attack. We characterize the conditions that fuzzy extractors need to satisfy to be secure, and present generic constructions from ordinary building blocks. As an illustration, we demonstrate how to use a biometric secret in a remote error tolerant authentication protocol that does not require any storage on the client's side.

1 Introduction

Often, one would like to be able to use some piece of cryptographic machinery, not with an exact, strictly random string as secret, but with an approximate, noisy rendition of it, which furthermore would not be perfectly random either. Such a "fuzzy" secret could be a measurement on a somewhat hidden biometric feature—a retinal scan rather than a thumbprint—, a long password imperfectly committed to memory, or even one's spontaneous answers to a list of subjective questions [EHMS00, FJ01]. Ideally, one would like to have a method to convert the above into as many cryptographically strong secrets usable for any purpose we like. A number of constructions geared toward specific applications have surfaced in the last few years [DFM98, JW99, MRW99, JS02]. Not surprisingly, related lines of work have also been pursued in different contexts, *e.g.*, for privacy amplification [BBCM95, BBR88], or for coping with noisy channels [Cre97].

The general idea is based on a two-step process, where an extraction function first transforms any sufficiently random fuzzy secret into an almost uniform random private string, and outputs some public information which is used in the regeneration step to reconstitute the exact same private string from a close enough approximation of the original fuzzy secret. Dodis *et al.* [DRS04] propose the most general definitions, and also introduce the notion of secure sketch (here renamed fuzzy sketch to avoid ambiguities), which works like an extractor except that no private string is extracted; rather, the goal is to allow an exact reconstruction of the original input given an approximation thereof. Although the repeated use of the regeneration function on many inputs is typically allowed, all these schemes implicitly assume that no more than a single extraction is ever performed from any secret—clearly a problematic state of affairs for biometric applications.

Toward a more robust definition of fuzzy sketch and extractor, we propose a security model based on the stringent notion of *adaptive chosen perturbation* attacks, wherein the adversary may query an oracle to perform extractions and regenerations based on chosen perturbations of the secret under attack. If the adversary is only given an extraction oracle, we speak of an outsider attack; in the general case we have an insider attack. We first show under the outsider security requirements how to achieve information theoretic security, and prove that certain existing constructions already satisfy these conditions. We then show how to harden the generic construction to withstand insider attacks, although in this case unconditional security is no longer feasible. We give fairly detailed security analysis based on simple assumptions, which we keep as general as possible to fit the generic nature of our constructions, and justify by showing their necessity; we rely on random oracles only in the case of extractors. Finally, we illustrate the power of our model by contructing a simple "zero storage" biometric authentication protocol based on universally reusable biometric certificates.

2 Preliminaries

We briefly recall a few classic notions needed in our constructions, mostly following [DRS04].

Metric Spaces. For the purpose of this paper, we define a metric space $\mathcal{M}$ as a finite set equipped with a non-negative integer distance function $\mathsf{d} : \mathcal{M} \times \mathcal{M} \to \mathbb{Z}_{\geq 0}$ which obeys the usual properties of a distance (symmetry, triangle inequality, zero distance between equal points). The elements of $\mathcal{M}$ are assumed to admit an efficient compact representation as bit strings of length $\mathcal{O}[\log_2 \#\mathcal{M}]$.

Hamming Distance. We usually consider multi-dimensional metric spaces of the form $\mathcal{M} = \Sigma^n$ for some alphabet Σ (usually a finite field $\mathbb{F}_p$), equipped with the Hamming distance. For any two words $w, w' \in \Sigma^n$, the Hamming metric $\mathsf{d}[w, w']$ is the number of coordinates in which they differ.

Error Correcting Codes. For a given choice of metric d, one can define error correcting codes in the corresponding space $\mathcal{M}$. A *code* is a subset $\mathcal{C} = \{w_1, ..., w_K\} \subseteq \mathcal{M}$. The set $\mathcal{C}$ is sometimes called *codebook*; its K elements are the *codewords*. The *(minimum) distance* of a code is the smallest distance d between two distinct codewords (according to the metric d). Given a codebook $\mathcal{C}$, we can define a pair of functions $\langle C, D\rangle$. The encoding function C is an injective map from the elements of some domain of size K to the elements of $\mathcal{C}$. The decoding function D maps any element $w \in \mathcal{M}$ to the pre-image $C^{-1}[w_k]$ of the codeword w_k that minimizes the distance $\mathsf{d}[w, w_k]$. The *error correcting distance* is the largest radius t such that for every element $w \in \mathcal{M}$ there is at most one codeword in the ball of radius t centered on w. For integer distance functions we have $t = \lfloor (d-1)/2 \rfloor$. A standard shorthand notation in coding theory is that of a $(\mathcal{M}, K, t)$-code.

We also define a complementary notion and say that the code has *error correction limit* t' if for any codeword $w_k \in \mathcal{C}$ and any element $w \in \mathcal{M}$ such that $C[D[w]] = w_k$ we have that $\mathsf{d}[w, w_k] \leq t'$.

Linear Codes. If the alphabet is a finite field $\Sigma = \mathbb{F}_p$ then $\mathcal{M} = \Sigma^n$ is a finite vector space. A *linear code* of parameters $[n, k, d]$ over $\mathbb{F}_p$ is a code whose codebook $\mathcal{C}$ is a vector subspace of $\mathbb{F}_p^n$—*i.e.*, $\mathcal{C}$ is closed under vector addition and scalar multiplication by elements of $\mathbb{F}_p$—such that $\mathcal{C}$ has size n^k and distance d. The natural notion of distance for linear codes is the Hamming metric.

The "square bracket" parameter notation $[n, k, d]$ is also used for non-linear codes over spaces of the form $\mathcal{M} = \Sigma^n$ when $k = \log_{\#\Sigma} \#\mathcal{C}$ is integral. Such a code is said to have *dimension* k.

Entropy. Let A and B be two random variables with values in the discrete domains $\mathcal{A}$ and $\mathcal{B}$. The *entropy* of A is defined as the expectation $\mathbf{H}[A] = \mathbf{E}_{a \xleftarrow{\$} A}[-\log_2 \mathbf{P}[A = a]]$. The *conditional entropy* of A given B is written $\mathbf{H}[A \mid B] = \mathbf{E}_{b \xleftarrow{\$} B}\mathbf{H}[A \mid B = b] = \mathbf{E}_{\langle a,b\rangle \xleftarrow{\$} \langle A,B\rangle}[-\log_2 \mathbf{P}[A = a \mid B = b]]$.

(Average) Min-Entropy. The notion of entropy quantifies the "expected randomness" of a random variable. To quantify the cryptographically more robust notion notion of "worst-case randomness", we consider the *min-entropy* of A which is defined as $\mathbf{H}_\infty[A] = -\log_2 \max_{a\in\mathcal{A}} \mathbf{P}[A = a]$. For conditional distributions, we use the notion of *average min-entropy*, which for A given B is defined as $\bar{\mathbf{H}}_\infty[A \mid B] = -\log_2 \mathbf{E}_{b \xleftarrow{\$} B}[\max_{a\in\mathcal{A}} \mathbf{P}[A = a \mid B = b]]$. This is not the expected min-entropy of A given B, but rather the (negative) logarithm of the average probability of the most likely value of A given B; it is more pessimistic since $\bar{\mathbf{H}}_\infty[A \mid B] \leq \mathbf{E}_{b \xleftarrow{\$} B}[\mathbf{H}_\infty[A \mid B = b]]$.

Statistical Distance. The *statistical distance* between two probability distributions A_1 and A_2 over a common discrete domain $\mathcal{A}$ is written $\mathbf{D}[A_1, A_2] = \frac{1}{2}\sum_{a\in\mathcal{A}} |\mathbf{P}[A_1 = a] - \mathbf{P}[A_2 = a]|$.

It is often useful to consider the statistical distance to a uniform distribution. We use the notation U_ℓ to denote a uniformly distributed random variable over $\{0, 1\}^\ell$.

Permutation Groups. Let $\mathcal{P} = \{\pi_p : \mathcal{M} \to \mathcal{M}\}$ be a family of functions indexed by p in some finite set. $\mathcal{P}$ is said to be a *permutation group* if $\langle \mathcal{P}, \circ\rangle$ is a group (observe that the π_p must be permutations of $\mathcal{M}$ since they have inverses in $\mathcal{P}$). The group operation $\circ$ in $\mathcal{P}$ and the action of the permutations π_p on $\mathcal{M}$ are implicitly assumed to be efficiently computable from canonical representations. We define the following properties of any such permutation group $\mathcal{P}$:

- $\mathcal{P}$ is *transitive* if for any pair of points $w, w' \in \mathcal{M}$, there is an (efficiently determinable) permutation $\pi_p \in \mathcal{P}$ such that $\pi_p[w] = w'$.
- $\mathcal{P}$ is *isometric* with respect to the distance d in $\mathcal{M}$ if for all permutation $\pi_p \in \mathcal{P}$ and points $w, w' \in \mathcal{M}$ it holds that $\mathsf{d}[\pi_p[w], \pi_p[w']] = \mathsf{d}[w, w']$.

3 Previous Notions Of Extractors

In this section, we review the definition of a fuzzy extractor as introduced by Dodis *et al.* [DRS04] and related notions. We then show by a counterexample that fuzzy extractors may be quite insecure if the same noisy secret is reused a few times.

3.1 Randomness Extractors

Intuitively, a (non-fuzzy) strong randomness extractor [NZ96] is a randomized function that tranforms its input from any biased distribution of sufficient min-entropy into an output that appears to be drawn from an almost uniform distribution. We require that this be the case even if one is given access to the random bits used by the extractor (but not its input).

Definition 1. An efficient (n, m', ℓ, ϵ)-*strong randomness extractor* (or *randomness extractor* for short) is a polynomial time randomized algorithm $\mathsf{Ext} : \{0, 1\}^n \to \{0, 1\}^\ell$ such that, for any random variable W over $\{0, 1\}^n$ with min-entropy m', it holds that $\mathbf{D}[\langle \mathsf{Ext}[W; R], R\rangle, \langle U_\ell, R\rangle] \leq \epsilon$. Here, $\mathsf{Ext}[W; R]$ denotes the application of Ext to the input word W using randomization bits R; the random variable R is required to have a uniform distribution independent from W and U_ℓ.

As shown in [RTS97], the theoretical limit is given by $\ell \leq m' - 2\,\log_2[1/\epsilon] + \mathcal{O}[1]$. A number of optimal constructions that also minimize the size of r are surveyed in [Sha02]. If the size of r is not critical, simpler optimal constructions can be obtained from pairwise independent hash functions [BBR88, HILL89].

3.2 From Fuzzy Sketches To Fuzzy Extractors

Dodis *et al.* [DRS04] define the following notions of fuzzy sketch (or secure sketch, in their terminology) and fuzzy extractor, and show how to construct the former can be transformed into the latter using a randomness extractor.

Definition 2. A $(\mathcal{M}, m, m', t)$-*fuzzy sketch* is a pair $\langle \mathsf{Fsk}, \mathsf{Cor} \rangle$ where:

Fsk is a (typically randomized) sketching function that on input $w \in \mathcal{M}$ outputs a *sketch* or redundancy data $\textsc{p} \in \{0,1\}^*$, such that for all random variable W over $\mathcal{M}$ with min-entropy $\mathbf{H}_\infty[W] \geq m$, the average min-entropy of W given $\mathsf{Fsk}[W]$ satisfies $\bar{\mathbf{H}}_\infty[W \mid \mathsf{Fsk}[W]] \geq m'$.

Cor is a correction function that given a word $w' \in \mathcal{M}$ and a sketch $\textsc{p}$ outputs a word $w'' \in \mathcal{M}$, such that for any $\textsc{p} \leftarrow \mathsf{Fsk}[w]$ and $\mathsf{d}[w, w'] \leq t$, it holds that $w'' = w$.

When we need to explicitly consider the random bits r used by Fsk on input w, we write $\mathsf{Fsk}[w; r]$. The functions Fsk and Cor are assumed efficiently computable, and the domain of r finite.

Definition 3. A $(\mathcal{M}, m, \ell, t, \epsilon)$-*fuzzy extractor* is a pair $\langle \mathsf{Gen}, \mathsf{Reg} \rangle$ where:

Gen is a (necessarily randomized) generation function that on input $w \in \mathcal{M}$ extracts a *private string* $\textsc{s} \in \{0,1\}^\ell$ and a *public string* $\textsc{q}$, such that for all random variable W over $\mathcal{M}$ such that $\mathbf{H}_\infty[W] \geq m$ and dependent variables $\langle \textsc{s}, \textsc{q} \rangle \leftarrow \mathsf{Gen}[W]$, it holds that $\mathbf{D}[\langle \textsc{s}, \textsc{q} \rangle, \langle U_\ell, \textsc{q} \rangle] \leq \epsilon$.

Reg is a regeneration function that given a word $w' \in \mathcal{M}$ and a public string $\textsc{q}$ outputs a string $\textsc{s}' \in \{0,1\}^\ell$, such that for any words $w, w' \in \mathcal{M}$ satisfying $\mathsf{d}[w, w'] \leq t$ and any possible pair $\langle \textsc{s}, \textsc{q} \rangle \leftarrow \mathsf{Gen}[w]$, it holds that $\textsc{s} = \mathsf{Reg}[w', \textsc{q}]$.

When we need to explicitly consider the random bits r used by Gen on input w, we write $\mathsf{Gen}[w; r]$. The functions Gen and Reg are assumed efficiently computable, and the domain of r finite.

Lemma 4 ([DRS04, Lemma 3.1]). *Let $\langle \mathsf{Fsk}, \mathsf{Cor} \rangle$ be a $(\mathcal{M}, m, m', t)$-fuzzy sketch. Suppose that Ext is a (n, m', ℓ, ϵ)-randomness extractor, assumed optimal and based on pairwise independent hashing so that $\ell = m' - 2\,\log_2[1/\epsilon]$. Then for uniformly distributed randomization strings r_1 and r_2, the following pair of algorithms $\langle \mathsf{Gen}, \mathsf{Reg} \rangle$ defines a $(\mathcal{M}, m, \ell, t, \epsilon)$-fuzzy extractor:*

$\mathsf{Gen}[w; \langle r_1, r_2 \rangle]$: *compute $\textsc{p} \leftarrow \mathsf{Fsk}[w; r_1]$ and $\textsc{s} \leftarrow \mathsf{Ext}[w; r_2]$, set $\textsc{q} \leftarrow \langle \textsc{p}, r_2 \rangle$, and output $\langle \textsc{s}, \textsc{q} \rangle$.*

$\mathsf{Reg}[w', \langle \textsc{p}, r_2 \rangle]$: *recover $w \leftarrow \mathsf{Cor}[w', \textsc{p}]$ and output $\textsc{s} \leftarrow \mathsf{Ext}[w; r_2]$.*

3.3 Concrete Constructions

Working towards showing a flaw in the above definitions, we recall for concreteness some fuzzy extractor constructions given in [DRS04].

Construction For Hamming Distance. A fuzzy extractor is easily obtained by viewing the notion of "fuzzy commitment" from [JW99] as a fuzzy sketch. We follow [DRS04, Section 4].

Let $C : \{0,1\}^k \to \{0,1\}^n$ be a (non-necessarily linear) binary code of parameters $[n, k, 2t+1]$, and let $D : \{0,1\}^n \to \{0,1\}^k$ be the matching decoding function. For random $r \in_\$ \{0,1\}^k$ we define the Juels-Wattenberg $(\mathcal{M}, m, m+k-n, t)$-fuzzy sketch over the Hamming space $\mathcal{M} = \{0,1\}^n$ as:

$$\mathsf{Fsk}[w; r] = w \oplus C[r] \ , \qquad\qquad \mathsf{Cor}[w', \mathrm{P}] = \mathrm{P} \oplus C[D[w' \oplus \mathrm{P}]] \ .$$

By combining the Juels-Wattenberg fuzzy sketch above with a randomness extractor as in Lemma 4, we immediately obtain a $(\mathcal{M}, m, \ell, t, \epsilon)$-fuzzy extractor $\langle \mathsf{Gen}, \mathsf{Reg} \rangle$ where $\ell = m + k - n - 2\ \log_2[1/\epsilon]$ and t measures Hamming distance. We call it the JW-DRS fuzzy extractor.

Permutation Based Extractors. Let $\mathcal{C} \subseteq \mathcal{M}$ be a code with encoding and decoding functions $\langle C, D \rangle$, and $\mathcal{P}$ a transitive group of isometric permutations in $\mathcal{M}$. Given such a family, a generic (randomized) "permutation based" fuzzy sketch $\langle \mathsf{Fsk}, \mathsf{Cor} \rangle$ is easily to construct:

$$\mathsf{Fsk}[w; r] = \mathrm{P} \quad \text{where} \quad \begin{cases} \bar{w} \leftarrow C[r] \in \mathcal{C} \\ \mathrm{P} \ \ \text{s.t.}\ \pi_{\mathrm{P}}[w] = \bar{w} \end{cases} \ , \qquad\qquad \mathsf{Cor}[w', \mathrm{P}] = (\pi_{\mathrm{P}}^{-1} \circ C \circ D \circ \pi_{\mathrm{P}})[w'] \ .$$

The principle is as follows. On input word w, the sketching function Fsk returns a permutation π_{P} that maps w to a randomly chosen codeword $\bar{w} \in \mathcal{C}$. Since the permutation is an isometry, the same permutation is used in the correction function Cor to turn any input w' in the vicinity of w into some word $\pi_{\mathrm{P}}[w']$ in the vicinity of $\bar{w}$; from there, the application of $C \circ D$ reconstitutes $\bar{w}$ and the subsequent inverse permutation π_{P}^{-1} maps it back to the original w. From there, the rest of the fuzzy extractor construction is as in Lemma 4. Dodis *et al.* [DRS04] show that if $\mathcal{C}$ is a $(\mathcal{M}, K, t)$-code and $\mathcal{P}$ is a transitive family of isometric permutations, the permutation based fuzzy sketch above is a $(\mathcal{M}, m, m', t)$-fuzzy sketch with entropy loss $m - m' = \log_2[\#\mathcal{P}] - \log_2[K]$, from which Lemma 4 gives a $(\mathcal{M}, m, \ell, t, \epsilon)$-fuzzy extractor of output size $\ell = m' - 2\ \log_2[1/\epsilon]$.

4 On The Insecure Reuse Of Fuzzy Extractors

Whereas Definitions 2 and 3 may be adequate for single-use fuzzy secrets, we now demonstrate various ways in which multiple invocations can coerce otherwise compliant fuzzy sketches and extractors to completely expose the secret. The avenues of attack we explore are: an insecure fuzzy sketch, a biased code, and a overly broad permutation family, respectively.

4.1 Fuzzy Sketch Indiscretion

Our first counterexample illustrates how a careless—yet compliant—fuzzy sketch and the extractor constructed from it can rapidly leak information about the input secret, if used multiple times.

A Flawed Construction. Let $\langle \mathsf{Fsk}, \mathsf{Cor} \rangle$ be a Juels-Wattenberg $(\mathcal{M}, m, m+k-n, t)$-fuzzy sketch as in Section 3.3. We construct a modified fuzzy sketch as follows:

$$\mathsf{Fsk}'[w; \langle r, r' \rangle] = \langle \mathsf{Fsk}[w; r], r', w \odot r' \rangle = \langle \mathrm{P}, r', b \rangle \ , \qquad\qquad \mathsf{Cor}'[w', \langle \mathrm{P}, r', b \rangle] = \mathsf{Cor}[w', \mathrm{P}] \ .$$

Here, $r \in_{\$} \{0,1\}^k$ and $r' \in_{\$} \{0,1\}^n$ are randomization strings assumed to be independently and uniformly distributed, and $b = w \odot r' \in \{0,1\}$ is the inner product of w and r'.

By the properties of $\langle \mathsf{Fsk}, \mathsf{Cor} \rangle$, for any random variable W of min-entropy m we know that $\bar{\mathbf{H}}_\infty[W \mid \mathsf{Fsk}[W]] \geq m + k - n$. Since r' is independent of W and b is a single bit, it follows that $\bar{\mathbf{H}}_\infty[W \mid \mathsf{Fsk}'[W]] \geq m + k - n - 1$. Thus, $\langle \mathsf{Fsk}', \mathsf{Cor}' \rangle$ is a $(\mathcal{M}, m, m+k-n-1, t)$-fuzzy sketch. We combine the fuzzy sketch $\langle \mathsf{Fsk}', \mathsf{Cor}' \rangle$ with a randomness extractor Ext as in Lemma 4, to yield a $(\mathcal{M}, m, \ell, t, \epsilon)$-fuzzy extractor $\langle \mathsf{Gen}', \mathsf{Reg}' \rangle$ with $\ell = m + k - n - 2\ \log_2[1/\epsilon] - 1$.

An Outsider Attack. We claim that the modified fuzzy extractor $\langle \mathsf{Gen}', \mathsf{Reg}' \rangle$ is flawed, though it is in all respects a "good" extractor according to the definition of [DRS04]. Indeed, assume that one makes a number q of independent calls to Gen' on the same (secret) input w^*. Assume for simplicity that $q \gg n$. Then, with high probability the q public strings $\mathrm{Q}'_1, ..., \mathrm{Q}'_q$ contain enough information to uniquely determine the secret word w^*. Furthermore, recovering w^* from that information amount to solving an (over-constrained) $n \times q$ linear system in $\mathbb{F}_2$, which can be done very efficiently. Once w^* is known, recovering the extracted private strings $\mathrm{S}_1, ..., \mathrm{S}_q$ is as easy as computing $\mathrm{S}'_i \leftarrow \mathsf{Reg}'[w^*, \mathrm{Q}'_i]$ for all $i \in \{1, ..., q\}$.

4.2 Coding Vulnerability

Improper fuzzy sketch constructions are not the only sources of information leaks. Even the *a priori* secure JW-DRS construction of Section 3.3 is prone to a total break when used with the wrong error correction code, if used multiple times. We outline the general argument. More details can be found in Appendix C.

Biased Codes. The argument is based on the notion of (non-linear) binary codes with a special property: on average over all the codewords in the codebook, the value 0 is more likely to appear than the value 1, at every coordinate of the code space. Specifically, we say that a p-ary $[n, k, d]$-code $\mathcal{C}$ has *bias* β, if, for a uniformly sampled random codeword $w \in_{\$} \mathcal{C}$, we have:

$$\forall i = 1, ..., n : \mathbf{P}[w|_i = 0] \geq \frac{1}{p} + \beta\ .$$

There are many ways to construct efficiently decodable biased codes. As an illustration, we refer to Appendix C for an explicit construction of such a code in the binary case. For now, we assume that $\mathcal{C}$ is a binary β-biased $[n, k, d]$-code with efficient encoding and decoding functions C and D.

When the JW-DRS construction of Section 3.3 is applied to the code $\langle C, D \rangle$, we obtain a $(\{0,1\}^n, m, \ell, t, \epsilon)$-fuzzy extractor $\langle \mathsf{Gen}, \mathsf{Reg} \rangle$ where $t = \lfloor (d-1)/2 \rfloor$ and $\ell = m + k - n - 2\ \log_2[1/\epsilon]$.

Majority Vote Attack. Recall that in the JW-DRS scheme the public string Q produced by a call to $\mathsf{Gen}[w^*]$ contains the substring $w^* \oplus C[r]$ for some r chosen uniformly at random. Since we are using a binary code with bias β, it follows that each bit of $w^* \oplus C[r]$ is equal to the corresponding bit of w^* with probability at least $\frac{1}{2} + \beta$. Thus, given a sufficiently large number $q = \Theta[\mathrm{poly}[1/\beta]]$ of public strings $\mathrm{Q}_1, ..., \mathrm{Q}_q$ derived from independent calls to $\mathsf{Gen}[w^*]$, it is indeed quite easy for an attacker to recover the secret w^* from public information: simply do a majority vote among all q public strings $\mathrm{Q}_1, ..., \mathrm{Q}_q$ for each of the n bits of $w^* \oplus C[r]$, one coordinate at a time.

4.3 Permutation Leaks

A third source of potential information leak can be found in the abstractions used in generic fuzzy sketches and extractors, such as the permutation based construction of Section 3.3. We show that a poor implementation of a particular abstraction can easily leak damaging information, if used multiple times.

Assume for the sake of illustration that $\mathcal{M}$ is the Hamming space $\mathbb{F}_p^n$ with vector addition $+$. Consider the permutation group $\mathcal{P} = \{\pi_p : w \mapsto p + w\} \cup \{\bar{\pi}_p : w \mapsto p - w\}$ consisting of all linear shifts (the π_p) and their mirror images (the $\bar{\pi}_p$). Clearly, $\mathcal{P}$ is a transitive isometric permutation group of size $\#\mathcal{P} = 2\,\#\mathcal{M}$, and it is easy to see that for any pair of words $w, \bar{w} \in \mathcal{M}$ there is exactly one "direct" and one "mirror" permutation in $\mathcal{P}$ maping w to $\bar{w}$, which we denote by $\pi_{w,\bar{w}}$ and $\bar{\pi}_{w,\bar{w}}$. Now, assume that $\langle \mathsf{Fsk}, \mathsf{Cor} \rangle$ is a permutation based $(\mathcal{M}, m, m', t)$-fuzzy sketch as in Section 3.3. The construction must specify how to select P s.t. $\pi_{\mathrm{P}}[w^*] = \bar{w}$ given a random $\bar{w} \in \mathcal{C}$. We specify it as follows: let $r' \leftarrow H[\pi_{w^*,\bar{w}}]$ for some fixed hash function H. If the parity of (a bit string representation of) $w^* \odot r'$ is 0, then pick P s.t. $\pi_{\mathrm{P}} = \pi_{w^*,\bar{w}}$; otherwise pick P s.t. $\pi_{\mathrm{P}} = \bar{\pi}_{w^*,\bar{w}}$.

In an attack, the adversary can easily determine whether P corresponds to $\pi_{w^*,\bar{w}}$ or $\bar{\pi}_{w^*,\bar{w}}$, and from there find the value of $w^* \odot r'$. If $w^* \odot r' = 0$, then $r' = H[\pi_{\mathrm{P}}]$ is easily recovered. Over q queries, an attacker can thus expect to obtain $q/2$ distinct P_i for which r'_i can be recovered this way. Given enough of these, it is easy to reconstruct the secret w^* using the method of Section 4.1.

This attack may seem contrived, but similar leaks can realistically occur in practice, *e.g.*, whenever P is selected deterministically among multiple choices from a set $\mathcal{P}$ that is ordered haphazardly. Although randomizing the choice of P would thwart this particular vulnerability, it is possible to mount much more powerful attacks in the same spirit if the adversary is allowed to obtain public strings for distinct secrets with a known or chosen relationship.

4.4 No Relief From Noisy Inputs

All the previous attacks assume that that multiple public strings are independently extracted from the same secret input. Since the secret is fuzzy, a more realistic scenario is to consider that the multiple extractions are performed on noisy variants of the fuzzy secret. We dispell the notion that such noise could somehow drastically hamper the above attacks.

Regarding the scheme of Section 4.2, observe that the attack is robust to small Hamming perturbations of the secret word w^*. Specifically, instead of Gen being applied multiple times to the same secret w^*, suppose that Gen is applied to q variations $w_1, ..., w_q$ of the secret w^*. It is easy to see that if all the w_i are contained within a ball of radius t centered on w^*, then the "majority vote" attack of Section 4.2 will produce a word $\tilde{w}$ that with high probability is also within distance t of the secret w^* (and possibly quite closer if the various perturbations cancel each other on average). From there, in virtue of the error tolerance that defines fuzzy extraction, the attacker can exactly regenerate the extracted private key strings $\mathrm{S}_1, ..., \mathrm{S}_q$ from the corresponding public strings $\mathrm{Q}_1, ..., \mathrm{Q}_q$, simply by computing $\mathrm{S}_i \leftarrow \mathsf{Reg}[\tilde{w}, \mathrm{Q}_i]$ for all $i \in \{1, ..., q\}$.

The attacks of Section 4.1 and 4.3 can also be adapted to cope with noisy secrets. Recall that in Section 4.1 we engineer fuzzy sketches that leak one bit of the input secret along a randomly chosen projection. Under noisy conditions, this results in an over-determined inconsistent set of contraints. The attacker can nonetheless attempt to solve, *e.g.*, for the least squared error approximation $\tilde{w}$, using techniques of linear algebra.

5 Secure Fuzzy Sketches And Extractors

The counterexamples of Section 4 clearly demonstrate the need for stronger notions of security for fuzzy sketches and extractors.

Our first notion is that of security against *outsider chosen perturbation attacks*; it directly addresses the vulnerabilities exposed in Section 4, and is mostly relevant to fuzzy sketches. In such attacks, the challenger holds a secret, and the adversary adaptively asks the challenger to run the sketching function Fsk on *chosen perturbations* of the secret—where a perturbation is a function specified by the adversary and applied by the challeger to the secret prior to processing a query. The adversary must not learn undue information about the secret from any number of such queries. (In the case of fuzzy extractors, the challenger runs Gen instead of Fsk, and shows the resulting public strings to the adversary, but not the private strings.)

Our second notion is that of security against *insider chosen perturbation attacks*; it is much more stringent and only applies to fuzzy extractors. In addition to making chosen perturbation queries on Gen as in the outsider attack, the adversary may adaptively ask the challenger to reconstruct certain private strings by applying Reg on chosen perturbations of the secret for arbitrary public strings (including ones from previous queries to Gen). The adversary must be computationally unable to recreate or distinguish any private string that it has not queried.

Perturbation Families. We need a manageable notion of perturbation that is useful to the adversary and manageable by the challenger. At the very least, perturbations should be efficiently computable. We keep the formal definition as simple and general as possible. Later, we will impose additional restrictions.

Definition 5. We call *perturbation* (the canonical representation of) any efficiently computable function $\delta : \mathcal{M} \to \mathcal{M}$. We call *perturbation family* any family $\Delta = \{\delta_d : \mathcal{M} \to \mathcal{M}\}$ of such functions, indexed by d in some finite set.

To fix ideas, suppose that $\mathcal{M}$ is a Hamming metric space, and define Δ as the set of all functions $f : \mathcal{M} \to \mathcal{M}$ such that $\forall w \in \mathcal{M}, \mathsf{d}[w, f[w]] \le \bar{d}$. In this case, the admissible perturbations are precisely the ones whose maximum displacement is bounded by $\bar{d}$; for example, the "shift" perturbations $\delta_d : \mathcal{M} \to \mathcal{M} : w \mapsto w + d$ are Δ-admissible provided that $\|d\| = \mathsf{d}[0, d] \le \bar{d}$.

In general, perturbations are not required to be invertible, or even composable in the sense that the composition of perturbations from a family may not itself be in the family.

5.1 Outsider Chosen Perturbation Security

Let Δ be a family of perturbations over some metric space $\mathcal{M}$ as previously defined. We define an adaptive outsider chosen perturbation attack against a fuzzy sketch (or a fuzzy extractor constructed from it) as the following game between a challenger and an adversary:

Preparation: The adversary sends to the challenger the specification (such as an efficient sampling procedure) of a random variable $W \in \mathcal{M}$.

Randomization: The challenger selects a secret word $w^* \in \mathcal{M}$ by randomly sampling W, and signals to the adversary that the query phase may begin.

Queries: The adversary presents arbitrarily many fuzzy sketching queries to the challenger. The queries are made adaptively, where for $k = 1, ...,$ the k-th query proceeds as follows.

The adversary chooses a perturbation $\delta_{d_k} \in \Delta$ and sends d_k to the challenger. The challenger runs Fsk on input word $w_k \leftarrow \delta_{d_k}[w^*]$ using fresh random bits r_k, obtaining a sketch $\mathrm{P}_k \leftarrow \mathsf{Fsk}[w_k; r_k]$, and responds to the query by giving P_k to the adversary.

Outcome: When the adversary decides that the queries are over, it produces a word $\hat{w^*} \in \mathcal{M}$. The winning condition for the adversary is that $\hat{w^*} = w^*$.

We call the unbounded adversary $\mathcal{A}_{\text{info}}$ in the above game a $\mathsf{Fuz\text{-}CPA}$ adversary.

Definition 6. Let $\langle \mathsf{Fsk}, \mathsf{Cor} \rangle$ be a $(\mathcal{M}, m, m', t)$-fuzzy sketch. If in the above game we have for all $\mathsf{Fuz\text{-}CPA}$ adversary whenever $\mathbf{H}_\infty[W] \geq m$ that $\mathbf{P}[\hat{w^*} = w^*] \leq 2^{-m'}$, then we say that the fuzzy sketch is unconditionally secure against adaptive *outsider* chosen perturbation attacks in Δ.

Outsider security for fuzzy extractors is defined in a similar way, except that the challenger responds to adversarial queries with the public output of Gen instead of the output of Fsk, and has to guess the private string corresponding to one of the public outputs it received. This corresponds to the game described in the coming section, where all private queries are disallowed.

5.2 Insider Chosen Perturbation Security

Let again Δ be a family of perturbations over some metric space $\mathcal{M}$ as previously defined. We define an adaptive insider chosen perturbation attack against a fuzzy extractor as the following game between a challenger and an adversary (which simultaneously describes a computational and a decisional version of the attack):

Preparation: The adversary specifies to the challenger a random variable $W \in \mathcal{M}$.

Randomization: The challenger randomly samples W to obtain a secret word $w^* \in \mathcal{M}$.

Public queries: The adversary presents up to q fuzzy generation queries to the challenger. The queries are made adaptively. For $i = 1, ..., q$, the i-th public query goes as follows. The adversary chooses a perturbation $\delta_{d_i} \in \Delta$ and sends d_i to the challenger. The challenger runs Gen on input word $w_i \leftarrow \delta_{d_i}[w^*]$ using fresh random bits r_i, obtaining a pair $\langle \mathrm{S}_i, \mathrm{Q}_i \rangle \leftarrow \mathsf{Gen}[w_i; r_i]$. The challenger discards the private string S_i, and responds to the query by giving the public string Q_i to the adversary.

Private queries: The adversary also presents up to q' fuzzy regeneration queries to the challenger. These queries are made adaptively and may be interspersed with public queries. For $j = 1, ..., q'$, the j-th private query goes as follows. The adversary chooses a perturbation $\delta_{d'_j} \in \Delta$ and a public string Q'_j, and sends both to the challenger. The challenger runs Reg on input word $w'_j \leftarrow \delta_{d'_j}[w^*]$ and public string Q'_j, obtaining a private string $\mathrm{S}'_j \leftarrow \mathsf{Reg}[w'_j, \mathrm{Q}'_j]$. The challenger responds by giving S'_j to the adversary.

Challenge: At some point, the adversary selects any public string $\hat{\mathrm{Q}} \in \{\mathrm{Q}_1, ..., \mathrm{Q}_q\}$ that was returned by the challenger in a previous public query, under the constraint that in any private query $\langle \delta, \hat{\mathrm{Q}} \rangle$ involving $\hat{\mathrm{Q}}$ the perturbation δ must have *minimum displacement* $\min_{w \in \mathcal{M}} \mathsf{d}[w, \delta[w]] > \bar{t}$. The adversary gives $\hat{\mathrm{Q}}$ to the challenger.

In the Decisional version only, the challenger then flips a fair coin $b \in_\$ \{0, 1\}$. If $b = 1$ it computes the corresponding private string $\mathsf{Reg}[w^*, \hat{\mathrm{Q}}]$ and gives it to the adversary, otherwise it draws a random string $\neq \mathsf{Reg}[w^*, \hat{\mathrm{Q}}]$ of equal length ℓ and returns it instead.

Additional queries: The adversary may make further public and private queries up to the respective quotas q and q'. An additional restriction is imposed that no private query $\langle \delta, \hat{\mathrm{Q}} \rangle$ be made on the challenge $\hat{\mathrm{Q}}$ unless δ has minimum displacement greater than $\bar{t}$.

Output: The adversary eventually outputs a private string candidate $\hat{\mathrm{s}}$. The winning condition for the adversary is that $\hat{\mathrm{s}} = \mathsf{Reg}[w^*, \hat{\mathrm{Q}}]$.

In the Decisional version, the adversary only outputs a single bit $\hat{b}$, and wins if $\hat{b} = b$.

We call the adversary $\mathcal{A}_{\text{comp}}$ in the computational game an OW-Fuz-CPA adversary[1]. For the decisional version, we refer to the adversary $\mathcal{A}_{\text{deci}}$ as an IND-Fuz-CPA adversary[2]. If ℓ is the size of the extracted private strings, we define each adversary's advantage in its respective game as:

$$\mathsf{Adv}_{\mathcal{A}_{\text{comp}}} = |\mathbf{P}[\hat{\mathrm{s}} = \mathrm{s}_{\hat{k}}] - 2^{-\ell}| \ , \qquad\qquad \mathsf{Adv}_{\mathcal{A}_{\text{deci}}} = |\mathbf{P}[\hat{b} = b] - \frac{1}{2}| \ .$$

Definition 7. Let $\langle \mathsf{Gen}, \mathsf{Reg} \rangle$ be a $(\mathcal{M}, m, \ell, t, \epsilon)$-fuzzy extractor. Let $\mathcal{A}$ be a (randomized) adversary for the (computational or decisional) game above, such that $\mathbf{H}_\infty[W] \geq m$ and all query perturbations are chosen from some family Δ. Suppose that $\mathcal{A}$ runs in time τ and makes q public and q' private queries, and that the private queries involving the challenge public string are further subject to the minimum displacement requirement $\min_{w \in \mathcal{M}} \mathsf{d}[w, \delta[w]] > \bar{t}$.

If for all such OW-Fuz-CPA adversary $\mathcal{A}$ we have $\mathsf{Adv}_{\mathcal{A}} \leq \alpha$, we say that the fuzzy extractor is $(\tau, q, q', \bar{t}, \alpha)$-one-way secure against adaptive *insider* chosen perturbation attacks in Δ.

If for all such IND-Fuz-CPA adversary $\mathcal{A}$ we have $\mathsf{Adv}_{\mathcal{A}} \leq \alpha$, we say that the fuzzy extractor is $(\tau, q, q', \bar{t}, \alpha)$-indistinguishable against adaptive *insider* chosen perturbation attacks in Δ.

Model Rationale. We require the challenge public string $\hat{\mathrm{Q}}$ to be one of the strings previously generated by the challenger, rather than any well-formed public string, since the point of the attack is to break a system under someone else's control, here represented by the challenger. Similarly, the adverary's objective is to guess the private string for the specific secret w^*, as opposed to, say, any perturbation thereof, since the point of the attack is to impersonate whomever the system was set up to protect or authenticate. Note that we could allow the target to be any small perturbation of the secret, but this would not substantially change the security properties thanks to error correction.

In the query phases however, the attacker is given much greater flexibility in its ability to probe and disturb the challenger using a wide range of perturbations and faulty inputs. This captures the idea of an adversary set out to "break into the system, by any means necessary".

Minimum Displacements. The reason for the minimum displacement restriction on challenge private queries is to ward against trivial queries that by design are intended to reveal the target private string, *e.g.*, $\langle \delta, \hat{\mathrm{Q}} \rangle$ for any δ whose maximum displacement is no greater than the error correction distance t. Incidentally we must take $\bar{t} \geq t$ for this to be of any use. The smaller the difference $\bar{t} - t$, the tighter the requirement, and the stronger the resulting security notion.

More generally, it is enough to require that the chosen perturbations for the relevant queries displace all but a negligible fraction of the points in $\mathcal{M}$ by a distance greater than $\bar{t}$ (as would, *e.g.*, a rotation about the origin). Specifically, the relaxed requirement asks that all perturbation δ used in a private query in conjunction with the challenge public string satisfy $\mathbf{P}[\mathsf{d}[W, \delta[W]] > \bar{t}] > 1 - 2^{-\ell}$

[1] OW-Fuz-CPA = one-wayness of fuzzy extraction against adaptive chosen perturbation attacks.

[2] IND-Fuz-CPA = indistinguishability of fuzzy extraction against adaptive chosen perturbation attacks.

for all random variable $W \in \mathcal{M}$ with minimum entropy $\mathbf{H}_\infty[W] \geq m$. To keep things simple, we stick with the previously stated definition.

5.3 An Alternative: Random Perturbation Security

Weaker forms of secure reusability can be achieved using relaxed security definitions. For instance, we can define the notion of a random perturbation attack. Here, instead of answering the queries using a perturbation function specified by the adversary to produce the perturbed secret w_i, the challenger would sample w_i from some distribution, possibly specified by the adversary, conditionally on the secret w^*. For instance, random perturbations could be distributed such that $\mathbf{P}[w_i \mid w^*]$ decreases exponentially with the distance $\mathsf{d}[w_i, w^*]$.

It may be argued that random perturbations are a plausible model of the physical reality of imperfect biometric measurements. However, it is not clear how appropriate it models the mental processes involved in the imperfect recall of a password—*e.g.*, if a user's secret is based on a list of favorite movies [JS02], the adversary could attempt to selectively distract her memory by playing movie themes in the computer room while she is entering her secret. In such circumstances, asking for chosen perturbation security may be erring on the side of caution.

Although this paper does not delve any further into this topic, the notion of security against random perturbations is worthy of further study.

6 Unconditional Outsider Security From Symmetric Subcodes

Our first general results show that unconditional outsider security can be achieved in a generic way from codes that feature sufficient "symmetry" with respect to the selected perturbation operator.

6.1 Fundamental Limitations

To temper one's optimism, we start by showing that no viable fuzzy extractor can withstand an active attack with unrestricted perturbations.

Admissible Perturbations. Suppose that the fuzzy sketch or extractor to break is non trivial, *i.e.*, there exist two words $w_1, w_2 \in \mathcal{M}$ on which it behaves differently. Then the adversary can recover any q-bit challenger secret w^* in only q public queries, using the following perturbation for query $k = 1, ..., q$:

$$\delta_{d_k} : w \mapsto \begin{cases} w_1 & \text{if } w|_k = 0 \\ w_2 & \text{if } w|_k = 1 \end{cases} ,$$

i.e., the k-th perturbation tests the k-th bit of its input and outputs w_1 or w_2 accordingly.

To avoid giving such an unfair advantage to the adversary, we need a reasonable notion of perturbation that treats all possible secret words in a comparable way. A natural solution is to require all perturbations to be *isometric permutations*. The theorems that follow in this section show that this is indeed a very natural notion of admissible perturbation.

6.2 Generic Construction

Our reusable fuzzy sketch construction is based on codes with certain symmetry properties, which we now define.

Weakly Symmetric Subcodes. We previously showed how to break fuzzy sketches and extractors by exploiting various asymmetries, *e.g.*, in the error correcting code or in the permutation family (in the case of a permutation based extractor). We need a notion of symmetry in order to close these loopholes. Since natural definitions of symmetry are based on groups of permutations, we define the following (very weak) notion of symmetry for a code $\mathcal{C}$ based on a permutation group.

Definition 8. Let $\mathcal{C}$ be a code in some finite space $\mathcal{M}$. Let $\mathcal{Q}$ be a group of permutations in $\mathcal{M}$. We say that an element $\omega_0 \in \mathcal{C}$ is a *$\mathcal{Q}$-pivot* of $\mathcal{C}$ if:

$$\forall \pi \in \mathcal{Q} : \pi[\omega_0] \in \mathcal{C} \ .$$

In other words, the set of images of ω_0 under the permutations in $\mathcal{Q}$ forms a subcode $\mathcal{C}' \subseteq \mathcal{C}$ closed under $\mathcal{Q}$ and on which $\mathcal{Q}$ acts transitively (*i.e.*, mapping any of its elements to any other). We emphasize that nothing is said about the effect of $\mathcal{Q}$ on the remainder of the code $\mathcal{C} \setminus \mathcal{C}'$.

A Generic Fuzzy Sketch. Equipped with the above notion of symmetry, we can construct a generic fuzzy sketch based on permutations that is unconditionally secure against outsider attacks.

Let $\mathcal{C}$ be a (not necessarily linear) code over a metric space $\mathcal{M}$. Let $\mathcal{P}$ be a transitive group of isometric permutations over $\mathcal{M}$. Suppose that $\mathcal{C}$ contains a $\mathcal{Q}$-pivot ω_0 where $\mathcal{Q}$ is some subgroup of $\mathcal{P}$. We define the generic fuzzy sketch $\langle \mathsf{Fsk}, \mathsf{Cor} \rangle$ as follows:

$$\mathsf{Fsk}[w; r] = \mathrm{P} \quad \text{where} \quad \begin{cases} p_1 \stackrel{r}{\leftarrow} \{p' : \pi_{p'} \in \mathcal{P}, \pi_{p'}[w] = \omega_0\} \\ p_2 \stackrel{r}{\leftarrow} \{p'' : \pi_{p''} \in \mathcal{Q}\} \\ \mathrm{P} \ \text{ s.t. } \pi_{\mathrm{P}} = \pi_{p_2} \circ \pi_{p_1} \in \mathcal{P} \end{cases} \qquad \mathsf{Cor}[w', \mathrm{P}] = (\pi_{\mathrm{P}}^{-1} \circ C \circ D \circ \pi_{\mathrm{P}})[w'] \ .$$

Here, the assignments $p_1 \stackrel{r}{\leftarrow} \{p'\}$ and $p_2 \stackrel{r}{\leftarrow} \{p''\}$ are randomized using different portions of r.

6.3 Information Theoretic Security

The following theorem relates the "entropy loss" achieved by the generic fuzzy sketch to the relative sizes of $\mathcal{P}$ and $\mathcal{Q}$. We see that the construction has an active (outsider) security comparable to the passive security of the permutation based construction of [DRS04], provided that the chosen code offers enough symmetry for the chosen family of perturbations.

Theorem 9. *Let $\mathcal{C} \subseteq \mathcal{M}$ be a $(\mathcal{M}, K, t)$-code in a finite metric space $\mathcal{M}$. Let $\mathcal{Q} \subseteq \mathcal{P}$ be a subgroup of a transitive isometric permutation group $\mathcal{P}$. Assume that the code $\mathcal{C}$ admits a $\mathcal{Q}$-pivot $\omega_0 \in \mathcal{C}$. Then the generic algorithms $\langle \mathsf{Fsk}, \mathsf{Cor} \rangle$ above form a $(\mathcal{M}, m, m', t)$-fuzzy sketch with unconditional security against adaptive outsider chosen perturbation attacks in any perturbation family $\Delta \subseteq \mathcal{P}$, provided that $m - m' \geq \log_2[\#\mathcal{P}] - \log_2[\#\mathcal{Q}]$.*

Proof. First, we show that the above construction is a fuzzy sketch with the required error correction capabilities. Specifically, we have the following.

Claim 9.1. $\langle \mathsf{Fsk}, \mathsf{Cor} \rangle$ is a fuzzy sketch with error correction distance $\geq t$ for all inputs in $\mathcal{M}$.

This already shows the security of the construction in the case of a single sketch or extraction.

Next, we bound the information that an adversary can obtain from repeated identical queries (*i.e.*, without perturbation). Consider the function $\overline{\mathsf{Fsk}} : w \mapsto \{\mathsf{Fsk}[w; r] : \forall r\}$ that maps any $w \in \mathcal{M}$ to the set of values taken by $\mathsf{Fsk}[w; r]$ for all possible random drawings of the hidden randomization parameter r. We successively obtain the following.

Claim 9.2. $\overline{\mathsf{Fsk}}[w]$ captures all information about w that can be gathered from an $\mathsf{Fsk}[w]$ oracle.

Claim 9.3. The map $\overline{\mathsf{Fsk}}$ defines a partition of $\mathcal{M}$ into n equivalence classes with $n \leq \#\mathcal{P}/\#\mathcal{Q}$.

Claim 9.4. The value of $\overline{\mathsf{Fsk}}[w^*]$ reveals at most $\log_2[\#\mathcal{P}/\#\mathcal{Q}]$ bits of information about w^*.

It follows that $\langle \mathsf{Fsk}, \mathsf{Cor} \rangle$ is a $(\mathcal{M}, m, m', t)$-fuzzy sketch for any $m - m' \geq \log_2[\#\mathcal{P}/\#\mathcal{Q}]$, which furthermore is unconditionally secure against repeated queries (*i.e.*, a "chosen perturbation" attack where the only perturbation available to the adversary is the identity map).

Last, we show that the ability to specify perturbations in $\Delta \subseteq \mathcal{P}$ does not provide additional information to the adversary. Precisely, we show that for any secret $w^* \in \mathcal{M}$, the challenger's answers to any (multi-)set of chosen perturbation queries in the family Δ do not collectively contain more information than $\overline{\mathsf{Fsk}}[w^*]$ itself. Using our previous claims, we find the following.

Claim 9.5. The value of $\overline{\mathsf{Fsk}}[\delta[w^*]]$ for any $w^* \in \mathcal{M}$ and $\delta \in \mathcal{P}$ is computable from $\overline{\mathsf{Fsk}}[w^*]$ and δ.

The security of the generic construction against adaptive chosen perturbation outsider attacks follows immediately from Claims 9.1, 9.2, 9.4, and 9.5. We refer to Appendix A.1 for detailed proofs of all the claims. $\square$

We then easily obtain an outsider secure fuzzy extractor using the construction of Lemma 4.

Corollary 10. *Under the assumptions of Theorem 9, there exists a $(\mathcal{M}, m, \ell, t, \epsilon)$-fuzzy extractor that is $(\infty, \infty, 0, 0, \alpha)$-IND-Fuz-CPA secure against adaptive chosen perturbation attacks in Δ, for arbitrary $\alpha > 0$, with $\ell = m + \log_2[\#\mathcal{Q}] - \log_2[\#\mathcal{P}] - 2\, \log_2[1/\epsilon]$.*

6.4 Generic Tightness

Our next theorem shows that the assumptions of Theorem 9 are "tight", in the sense that if there exists any fuzzy sketch (not necessarily based on permutations) with outsider security *vs.* a sufficiently powerful perturbation family, then we necessarily have all the elements we had to assume for the generic fuzzy sketch construction to go through.

This theorem serves to show that the requirements from the results of the previous section are far from being arbitrary.

Theorem 11. *Assume that $\langle \mathsf{Fsk}, \mathsf{Cor} \rangle$ is a $(\mathcal{M}, m, m', t)$-fuzzy sketch unconditionally secure against adaptive outsider chosen perturbation attacks in a family Δ (containing the identity perturbation). Suppose that a subset $\Delta' \subseteq \Delta$ generates a transitive group $\mathcal{P}$ of isometric permutations in $\mathcal{M}$. Then there exists a subgroup $\mathcal{Q} \subseteq \mathcal{P}$ and a $(\mathcal{M}, K, t)$-code $\mathcal{C} \subseteq \mathcal{M}$ that contains a $\mathcal{Q}$-pivot $\omega_0 \in \mathcal{C}$ with $K = \#\mathcal{M}\,\#\mathcal{Q}/\#\mathcal{P}$, where furthermore $m - m' \geq \log_2[\#\mathcal{P}] - \log_2[\#\mathcal{Q}]$.*

Proof. Since the challenger responses may be randomized, we start by deterministically characterizing the information that an unbounded adversary may gather in an outsider attack.

1. We define the function $\overline{\mathsf{Fsk}} : w \mapsto \{\mathsf{Fsk}[w; r] : \forall r\}$ that maps any element $w \in \mathcal{M}$ to the set of all possible randomized values of $\mathsf{Fsk}[w]$. The function $\overline{\mathsf{Fsk}}$ is effectively computable with arbitrarily high probability given a black box simulator for Fsk, since with enough queries one will eventually exhaust the finite set of possible randomization strings used by Fsk.

2. We define the function $\overline{\overline{\mathsf{Fsk}}} : w \mapsto \{\langle \delta, \overline{\mathsf{Fsk}}[\delta[w]] \rangle : \forall \delta \in \Delta\}$ that maps any element $w \in \mathcal{M}$ to the relation between the admissible perturbations $\delta \in \Delta$ and the values taken by $\overline{\mathsf{Fsk}}$ on the perturbated input $\delta[w]$. Since the set Δ is finite, this function can be computed from $\overline{\mathsf{Fsk}}$.

We now successively show the following claims.

Claim 11.1. Given a randomized oracle for $\mathsf{Fsk}[\delta[w^*]]$ for chosen $\delta \in \Delta$, the value of w^* can be disambiguated with arbitrarily high probability up to the set of preimages of $\overline{\overline{\mathsf{Fsk}}}[w^*]$.

Claim 11.2. Conversely, the value of $\overline{\overline{\mathsf{Fsk}}}[w^*]$ captures the total information about w^* that can be gathered from arbitrarily many queries to $\mathsf{Fsk}[\delta[w^*]]$ for chosen perturbation in Δ.

Claim 11.3. The number n of equivalence classes induced over $\mathcal{M}$ by $\overline{\overline{\mathsf{Fsk}}}$ is bounded as $n \leq 2^{m-m'}$, unless $m' = 0$ (in which case the theorem is vacuous).

Claim 11.4. There is a subgroup $\mathcal{Q} \subseteq \mathcal{P}$ that preserves the equivalence structure between classes, where $\log_2[\#\mathcal{P}] - \log_2[\#\mathcal{Q}] = \log_2[n] \leq m - m'$.

Claim 11.5. Each equivalence class $\mathcal{C}_i$ forms a $(\mathcal{M}, K, t)$-code of size $K = \#\mathcal{M}/n$, whose elements are all $\mathcal{Q}$-pivots of $\mathcal{C}_i$.

The theorem follows from Claims 11.4 and 11.5. We refer to Appendix A.2 for detailed proofs of all the claims. □

6.5 Example: Linear Codes In Hamming Spaces

Let $\mathcal{M}$ be the n-dimensional vector space $\mathbb{F}_p^n$ with the Hamming metric $\mathsf{d} : \mathcal{M} \times \mathcal{M} \to \{0, 1, ..., n\}$, and suppose that $\mathcal{C} \subseteq \mathcal{M}$ is a linear p-ary $[n, k, d]$-code in that space. Let $\mathcal{P}$ be the transitive isometric permutation group of all maps $\pi_p : \mathcal{M} \to \mathcal{M} : w \mapsto w + p$ for $p \in \mathcal{M}$. Let $\mathcal{Q}$ be the subset of maps $\pi_p \in \mathcal{P}$ such that $p \in \mathcal{C}$. Since the code $\mathcal{C}$ is linear, it is easy to see that $\mathcal{Q}$ is closed under function inversion and function composition; $\mathcal{Q}$ is thus a subgroup of $\mathcal{P}$, and any element $\omega_0 \in \mathcal{C}$ is a $\mathcal{Q}$-pivot of $\mathcal{C}$. We have $\#\mathcal{P} = \#\mathcal{M} = p^n$ and $\#\mathcal{Q} = \#\mathcal{C} = p^k$. By Theorem 9 the generic construction of Section 6.2 immediately gives us a $(\mathcal{M}, m, m', t)$-fuzzy sketch unconditionally secure against outsider attacks with $t = \lfloor (d-1)/2 \rfloor$ provided that $m - m' \geq (\log_2 p)\,(n-k)$. By Corollary 10 we get a $(\mathcal{M}, m, \ell, t, \epsilon)$-fuzzy extractor unconditionally secure against outsider attacks with binary output size $\ell = m' - 2\,\log_2[1/\epsilon] \leq m - (\log_2 p)\,(n-k) - 2\,\log_2[1/\epsilon]$.

In the binary case $\mathbb{F}_p = \mathbb{F}_2$, it is easy to show that this construction precisely reduces to the JW-DRS fuzzy extractor previously mentioned. This proves that the JW-DRS construction is unconditionally secure against outsider attacks provided that it is used with a linear code.

We note that the use of linear codes for reconstructing imperfectly shared secret information has been extensively studied, *e.g.*, in the context of privacy amplification and information reconciliation [BBR88, BBCM95]. It has also been observed in [DRS04] that the fuzzy commitment scheme of [JW99] described in Section 3.3 reduces when applied on a linear code to a deterministic fuzzy sketch equal to the code's *syndrome* function $S : \mathcal{M} \to \mathcal{M} : w \mapsto (w - C[D[w]])$.

6.6 Counterexample: Text Edit Distance

Dodis *et al.* [DRS04] also present a fuzzy sketch construction for text, based on the notion of edit distance. Roughly speaking, the edit distance between two texts A and B is the length of an "edition script" that turns A into B using combinations of three basic commands: insertion, deletion, and displacement of sequences of characters at specified locations in the text.

A natural choice for the family of perturbations is the set of all edition scripts, possibly with a length restriction. Unfortunately, it is easy to see that this gives too much power to the adversary.

Consider the script $Subst[char, pos]$ that substitutes the supplied character $char$ for the character at the given position pos in the text. It is easy to see that the instantiation of this script for specific $char$ and pos gives a perturbation that allows the adversary to test whether the pos-th character in the secret text is equal to $char$. This allows the adversary to quickly recover the hidden secret text character by character, in a similar way as in the example given in Section 6.1.

7 Ideal Insider Security Through Random Hashing

We now convert an unconditionally outsider secure fuzzy sketch, such as the one of the previous section, into a fuzzy extractor with insider security using random oracles [BR93]. The security is no longer unconditional, but is "ideal" in the sense that all incurred security losses result either from random collisions or from the adversary's ability to defeat the randomness assumption.

Recall how in our previous construction we arranged to confine all public queries to a symmetric subcode $\mathcal{C}' \subseteq \mathcal{C}$, steering clear from any potentially recognizable "landmark" lurking in $\mathcal{C} \setminus \mathcal{C}'$. Unfortunately, private queries cannot be confined so easily, as a clever query $\langle \delta, \text{Q} \rangle$ can always cause any secret w^* to be corrected to any codeword in $\mathcal{C}$, not just $\mathcal{C}'$. However, we can randomly shuffle things around at decoding time to render all codewords indistinguishable up to permutations in $\mathcal{Q}$, thereby preventing too much information from being leaked. Nevertheless, by the $\mathcal{Q}$-symmetry of the subcode $\mathcal{C}'$, any legitimate query that only involves codewords in the subcode will be impervious to the randomization. See Appendix B for a more detailed explanation.

Let thus $\omega_0 \in \mathcal{C} \subseteq \mathcal{M}$ and $\mathcal{Q} \subseteq \mathcal{P}$ be as in Section 6. First, we define the fully randomized generic fuzzy sketch $\langle \mathsf{Fsk}, \mathsf{Cor} \rangle$ as follows:

$$\mathsf{Fsk}[w; r] = \text{P} \quad \begin{cases} p_1 \xleftarrow{r} \{p' : \pi_{p'} \in \mathcal{P}, \pi_{p'}[w] = \omega_0\} \\ p_2 \xleftarrow{r} \{p'' : \pi_{p''} \in \mathcal{Q}\} \\ \text{P} \;\; \text{s.t. } \pi_\text{P} = \pi_{p_2} \circ \pi_{p_1} \in \mathcal{P} \end{cases} \qquad \begin{array}{r} \mathsf{Cor}[w', \text{P}] = (\pi_\text{P}^{-1} \circ \pi^{-1} \circ C \circ D \circ \pi \circ \pi_\text{P})[w'] \\ \text{for random } \;\; \pi \leftarrow \mathcal{Q} \; . \end{array}$$

Again, the assignments $p_1 \xleftarrow{r} \{p'\}$ and $p_2 \xleftarrow{r} \{p''\}$ are randomized using different portions of r.

Next, assuming a random oracle H, we define the full generic fuzzy extractor $\langle \mathsf{Gen}, \mathsf{Reg} \rangle$ as follows:

$$\mathsf{Gen}[w; \langle r, r' \rangle] = \langle \text{S}, \text{Q} \rangle \quad \text{where} \quad \begin{cases} \text{P} = \mathsf{Fsk}[w; r] \\ \text{S} = \mathsf{H}[w, r', \text{P}] \\ \text{Q} = \langle \text{P}, r' \rangle \end{cases} \qquad \begin{array}{r} \mathsf{Reg}[w', \text{Q}] = \mathsf{H}[\mathsf{Cor}[w', \text{P}], r', \text{P}] \\ \text{where } \;\; \langle \text{P}, r' \rangle = \text{Q} \; . \end{array}$$

Here, H is a hash function treated as a random oracle in the analysis, with inputs in $\mathcal{M} \times \{0,1\}^{\ell'} \times \{0,1\}^{\ell''}$ and outputs in $\{0,1\}^{\ell}$. We assume that the random input r' is drawn from some $\{0,1\}^{\ell'}$ and that the representation of the fuzzy sketch P fits in $\{0,1\}^{\ell''}$.

Notice that both functions Fsk and Cor are now randomized, and thus so are Gen and Reg.

Theorem 12. *Under the conditions of Theorem 9 where the code $\mathcal{C}$ has error correction limit $\leq \bar{t}$, the algorithms $\langle \mathsf{Gen}, \mathsf{Reg} \rangle$ constitute a $(\mathcal{M}, m, \ell, t, \epsilon)$-fuzzy extractor, for any $\epsilon \geq \sqrt{2^{\ell - m'}}$, that is $(\infty, \infty, q', \bar{t}, \alpha)$-OW-Fuz-CPA and IND-Fuz-CPA secure whenever $\alpha \geq \binom{q'}{2} 2^{-\ell} + q' \, 2^{-m'} + \epsilon$, in the random oracle model, where q' also includes direct queries to the random oracle.*

Proof. The stated bound on ϵ follows immediately from the fact that a random oracle is an optimal (n, m', ℓ, ϵ)-randomness extractor, and thus $m' = \ell - 2\log_2[\epsilon] + \mathcal{O}[1]$ per [RTS97].

The rest of the proof is an information theoretic argument that bounds the knowledge available to the adversary under the stated randomization of Reg. See Appendix B for details. ☐

It is currently an open problem to achieve $\mathsf{OW\text{-}Fuz\text{-}CPA}$ security without random oracles.

8 "Zero Storage" Remote Biometric Authentication

To demonstrate the power of the reusable fuzzy extractor machinery, we briefly present a remote biometric authentication protocol with third party certification, that does not require Alice to securely or insecurely store anything—other than her fuzzy secret.

Suppose that Alice wishes to remotely authenticate herself to Bob using biometrics. Due to privacy concerns, she does not wish to reveal any of them to Bob (even if he does not play the protocol by the rules, and/or colludes with other Bobs against her). Conversely, for the authentication to be meaningful, Bob wants some assurance that Alice is in fact in possession of her purported biometrics at the time the authentication is taking place (*i.e.*, that nobody is impersonating her). We assume that there is a third party, Trent, whom Bob trusts to honestly certify Alice's biometrics, and to whom Alice will temporarily grant access to her biometrics for the purpose of generating such a certificate. Alice will want to be able to obtain as many or as few of those certificates as she wants, and to reuse as many of them with multiple Bobs, some of whom may be dishonest, without fearing privacy leaks or risking impersonation. The protocol is as follows.

Certification: Under Trent's supervision, and using Alice's own secret biometrics w^*:

1. Alice generates a random string pair $\langle \mathrm{S}, \mathrm{Q} \rangle \leftarrow \mathsf{Gen}[w^*]$ using an insider secure fuzzy extractor as that of Section 7;
2. Alice derives the public key $\mathsf{pbk}_{\mathrm{S}}$ that corresponds to the private string S viewed as a private key in some existentially unforgeable ($\mathsf{UF\text{-}CMA}$) signature scheme $\langle \mathsf{Sign}, \mathsf{Verify} \rangle$. (If S is not a legitimate private key, one is deterministically derived from it first).

If Trent is satisfied that Alice has executed the steps honestly, he certifies the binding between Alice's name and the public key $\mathsf{pbk}_{\mathrm{S}}$, *i.e.*, he signs the pair $\langle \text{"Alice"}, \mathsf{pbk}_{\mathrm{S}} \rangle$. In the sequel, we take $\mathsf{pbk}_{\mathrm{S}}$ to denote the public key accompanied with its certificate.

At this point, Alice may send the pair $\langle \mathrm{Q}, \mathsf{pbk}_{\mathrm{S}} \rangle$ to Bob, or even publish it for everyone to see.

Challenge: At any time when appropriate (*e.g.*, whenever Alice desires to authenticate to Bob), Bob sends Alice a fresh random challenge c_{nonce} and reminds her of her public string Q.

Response: Using what Bob claims to be her public string Q, and an approximation of her fuzzy secret biometrics $\tilde{w^*}$, Alice responds to the challenge as follows:

1. Alice recovers her private string $\tilde{\mathrm{S}} \leftarrow \mathsf{Reg}[\tilde{w^*}, \mathrm{Q}]$;
2. Alice signs the challenge and gives Bob the signature $s_{nonce} \leftarrow \mathsf{Sign}[\tilde{\mathrm{S}} : c_{nonce}]$.

Verification: Bob authenticates Alice by checking the validity of the signature under her authentic public key $\mathsf{pbk}_{\mathrm{S}}$, *viz.*, evaluating $\mathsf{Verify}[\mathsf{pbk}_{\mathrm{S}} : c_{nonce}, s_{nonce}]$.

Other black box identification schemes can be substituted for the last three steps.

The important point is that the protocol does not require Alice to "remember" anything other than her fuzzy secret (and in particular does not have to obtain Trent's authentic public key to verify a certificate). Alice's credentials remain secure in an attack where Alice is given corrupted Q by a malicious Bob.

Security Analysis. The protocol passes muster with Bob in that it properly authenticates Alice. Indeed, since the signatures are existentially unforgeable, we have non-repudiation, and, thus, knowledge of the private key is required to properly respond to a new challenge.

The protocol is also to Alice's taste in terms of protection of her privacy, at least against a computationally bounded adversary. Regarding certification, since the signature scheme is secure, we know that neither $\mathsf{pbk}_{\mathrm{S}}$ nor the signatures created from S computationally reveal anything about the private string S. Thus, in the adversary's view, each instance of the certification phase reduces to nothing more than a public query in the insider game of Section 5.2. Regarding the authentication handshakes, there are two cases to consider: whether Bob honestly or dishonestly "reminds" Alice of her public string Q. In the honest case, Alice's responses to Bob are safe, since when she uses the correct Q she creates a signature under her genuine private key S, which as we noted earlier does not computationally leak anything about S. In the dishonest case, the fuzzy extractor's insider security property ensures that, for any bogus public string $\mathrm{Q}' \neq \mathrm{Q}$ of Bob's crafting, Alice will not leak any computational knowledge about S by recreating an (incorrect) private string S' using Q'. Signatures generated from S' are *a fortiori* devoid of useful information.

The above properties continue to hold if Alice uses the same certificate with multiple Bobs, or conversely obtains multiple certificates and uses them with the same correspondent.

Related Key Attacks. Observe that we need a fuzzy extractor with insider security for the following (rather counter-intuitive) reason: although we know that issuing signatures under UF-CMA signature scheme does not computationally leak the private key, we cannot assume that this remains the case when signatures are also issued under other, related private keys. If the signature is well behaved in this respect then a (suitably defined) outsider secure fuzzy extractor suffices for this application.

9 Conclusion

We have studied the question of generating keys of cryptographic quality from non uniformly distributed, non perfectly reproducible "fuzzy" processes, focusing on the notions of fuzzy sketches and fuzzy extractors. Dealing with fuzzy secrets is a problem of great practical significance in applications where security relies at least in part on fuzzy secrets such as biometric measurements or imperfectly memorized passwords.

We demonstrated with a number of simple attacks that the existing definitions and constructions are inadequate and may lead to a total break of security in any circumstance where one is compelled to reuse the same fuzzy secret—which severely undermines their adequacy for biometrics.

We introduced two strong security models for fuzzy sketches and extractors that allow reusable secrets; in the first model the adversary is an outsider, and the other in which it is an insider. Our models are based on the security notion of "chosen perturbation attack".

We presented generic outsider secure fuzzy sketch and extractor constructions, and precisely characterized the conditions under which information theoretic security can be achieved.

We then extended our method to handle the case of insider attacks, and showed how to transform any outsider secure fuzzy sketch into an insider secure fuzzy extractor using random oracles.

We finally illustrated the power of our model with a simple zero storage fuzzy authentication protocol that remains secure even if the secret holder is unable or unwilling to remember anything but her fuzzy secret.

Acknowledgements

The author thanks Yevgeniy Dodis and Jonathan Katz for insightful comments.

References

[BBCM95] C.H. Bennett, G. Brassard, C. Crépeau, and U. Maurer. Generalized privacy amplification. *IEEE Trans. Information Theory*, 41(6):1915–1923, 1995.

[BBR88] C.H. Bennett, G. Brassard, and J. Robert. Privacy amplification by public discussion. *SIAM J. Computing*, 17(2):210–229, 1988.

[BR93] M. Bellare and P. Rogaway. Random oracle are practical: A paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security—CCS 1993*, pages 62–73, 1993.

[Cre97] C. Crepeau. Efficient cryptographic protocols based on noisy channels. In *Proc. Advances in Cryptology—Eurocrypt '97*, pages 306–317, 1997.

[DFM98] G. Davida, Y. Frankel, and B. Matt. On enabling secure applications through offline biometric identification. In *Proc. IEEE Symp. Security and Privacy*, pages 148–157, 1998.

[DRS03] Y. Dodis, L. Reyzin, and A. Smith. Fuzzy extractors and cryptography, or how to use your fingerprints. Cryptology ePrint Archive, Report 2003/235, 2003. `http://eprint.iacr.org/`.

[DRS04] Y. Dodis, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *Proc. Advances in Cryptology—Eurocrypt '04*, 2004. Full paper available as [DRS03].

[EHMS00] C. Ellison, C. Hall, R. Milbert, and B. Schneier. Protecting keys with personal entropy. *Future Generation Computer Systems*, 16:311–318, 2000.

[FJ01] N. Frykholm and A. Juels. Error-tolerant password recovery. In *Proc. ACM Conf. Computer and Communications Security*, pages 1–8, 2001.

[HILL89] J. Håstad, R. Impagliazzo, L. Levin, and M. Luby. A pseudorandom generator from any one-way function. In *Proc. 21st ACM Symp. Theory of Computing*, 1989.

[JS02] A. Juels and M. Sudan. A fuzzy vault scheme. In *IEEE Int. Symp. Information Theory*, 2002.

[JW99] A. Juels and M. Wattenberg. A fuzzy commitment scheme. In *Proc. ACM Conf. Computer and Communications Security*, pages 28–36, 1999.

[MRW99] F. Monrose, M. Reiter, and S. Wetzel. Password hardening based on keystroke dynamics. In *Proc. ACM Conf. Computer and Communications Security*, pages 73–82, 1999.

[NZ96] N. Nisan and D. Zuckerman. Randomness is linear in space. *JCSS*, 52(1):43–52, 1996.

[RTS97] J. Radhakrishnan and A. Ta-Shma. Tight bounds for depth-two superconcentrators. In *Proc. 38th IEEE Symp. Foundations of Computer Science*, pages 585–594, 1997.

[Sha02] R. Shaltiel. Recent developments in explicit constructions of extractors. *Bul. EATCS*, 77:67–95, 2002.

A Unconditional Outsider Security Proofs

We give detailed proofs of the various claims used in the two main theorems of Section 6.

Remark. At this juncture, we should mention that all proofs to follow proceed under the simplifying assumptions that Fsk uses uniformly distributed random bits r and that $\mathsf{Fsk}[w;r]$ takes different values for distinct r. If this is not the case, $\overline{\mathsf{Fsk}}[w]$ can be extended to store relative frequency information and/or generalized into a multiset of values taken by $\mathsf{Fsk}[w;r]$. Modulo some technicalities to account for this extra information, all the proofs would remain essentially unchanged.

A.1 Proving Theorem 9 (Information Theoretic Security)

Proof of Claim 9.1. Since the pivot ω_0 and all its images under $\mathcal{Q}$ are codewords in $\mathcal{C}$, we know that $\mathsf{Fsk}[w]$ on any input w returns a permutation π_{P} such that $\pi_{\mathrm{P}}[w] \in \mathcal{C}$. Since $\pi_{\mathrm{P}} \in \mathcal{P}$ is isometric, for any w' such that $\mathsf{d}[w, w'] \leq t$ we also have $\mathsf{d}[\pi_{\mathrm{P}}[w], \pi_{\mathrm{P}}[w']] \leq t$. Since the code $\mathcal{C}$ has an error correcting distance $\geq t$ it follows that $(C \circ D \circ \pi_{\mathrm{P}})[w'] = \pi_{\mathrm{P}}[w]$. Consequently, $\mathsf{Cor}[\cdot, \pi_{\mathrm{P}}]$ corrects w' to w', as required. $\square$

Proof of Claim 9.2. The $\mathsf{Fsk}[w]$ oracle can be simulated without knowing w by returning randomly selected elements from the set $\overline{\mathsf{Fsk}}[w]$. The claim follows. $\square$

Proof of Claim 9.3. To start, observe that $\overline{\mathsf{Fsk}}[w] = \{\pi_p \in \mathcal{P} : \exists \pi' \in \mathcal{Q}, \pi_p[w] = \pi'[\omega_0]\}$, *viz.*, the set of permutations π_p that map w to any image of ω_0 under $\mathcal{Q}$. Indeed, the inclusion "$\subseteq$" is immediate. To see the converse inclusion "$\supseteq$", observe that for all such π_p we have $({\pi'}^{-1} \circ \pi_p)[w] = \omega_0$ for some $\pi' \in \mathcal{Q}$, which means that "$\pi_{p_2} \circ \pi_{p_1}$" $\leftarrow \pi' \circ ({\pi'}^{-1} \circ \pi_p)$ is a legitimate output of $\mathsf{Fsk}[w]$.

We bound the number n of distinct images of $\overline{\mathsf{Fsk}}$ on all possible inputs $w \in \mathcal{M}$ as follows. Consider the images $\overline{\mathsf{Fsk}}[w_1], \overline{\mathsf{Fsk}}[w_2] \subseteq \mathcal{P}$ of any two points $w_1, w_2 \in \mathcal{M}$. By transitivity of $\mathcal{P}$, we can fix a permutation $\pi_{2\to1} : w_2 \mapsto w_1$ in $\mathcal{P}$ that turns all permutations $\pi \in \overline{\mathsf{Fsk}}[w_1]$ into permutations $\pi' = (\pi \circ \pi_{2\to1}) \in \overline{\mathsf{Fsk}}[w_2]$. Since this operation is reversible (using $(\pi_{2\to1})^{-1}$), it follows that $\pi_{2\to1}$ defines an "action" in the space $\mathcal{F} = \{\overline{\mathsf{Fsk}}[w] : w \in \mathcal{M}\}$ which transforms $\overline{\mathsf{Fsk}}[w_1]$ into $\overline{\mathsf{Fsk}}[w_2]$. It is easy to see that each permutation in $\mathcal{P}$ defines a corresponding action in $\mathcal{F}$, where furthermore the actions in $\mathcal{F}$ compose according to the group law of $\mathcal{P}$. In other words, $\mathcal{P}$ defines a transformation group over the space $\mathcal{F} = \{\overline{\mathsf{Fsk}}[w] : w \in \mathcal{M}\}$.

Since $\mathcal{P}$ is transitive, the group orbit (*i.e.*, the set of reachable points) from any fixed element of $\mathcal{F}$ under the transformations in $\mathcal{P}$ is the entire space $\mathcal{F}$. Thus, the action of the permutations in $\mathcal{P}$ on the chosen fixed element defines a surjection f from $\mathcal{P}$ to $\mathcal{F}$. Since $\mathcal{Q}$ is a subgroup of $\mathcal{P}$, the subsets $\overline{\mathsf{Fsk}}[w] \subseteq \mathcal{P}$ are invariant with respect to left composition by $\mathcal{Q}$, *i.e.*, we have $\mathcal{Q} \circ \overline{\mathsf{Fsk}}[w] = \overline{\mathsf{Fsk}}[w]$ where $\mathcal{Q} \circ \overline{\mathsf{Fsk}}[w]$ denotes the set $\{(\pi' \circ \pi) : \pi' \in \mathcal{Q}, \pi \in \overline{\mathsf{Fsk}}[w]\} \subseteq \mathcal{P}$. This implies that f maps at least $\#\mathcal{Q}$ distinct elements of $\mathcal{P}$ to each element of $\mathcal{F}$, which requires that there be no more than $\#\mathcal{P}/\#\mathcal{Q}$ elements in $\mathcal{F}$. Accordingly, the map $\overline{\mathsf{Fsk}} : \mathcal{M} \to \mathcal{F}$ defines a partition of $\mathcal{M}$ into $n \leq \#\mathcal{P}/\#\mathcal{Q}$ equivalence classes. $\square$

Proof of Claim 9.4. We now compute a uniform upper bound on the information content of $\overline{\mathsf{Fsk}}[w]$ for arbitrary $w \in \mathcal{M}$. We know that the function $\overline{\mathsf{Fsk}}$ partitions $\mathcal{M}$ into n equivalence classes $\mathcal{C}_1, ..., \mathcal{C}_n \subseteq \mathcal{M}$, where each class is defined by the common value taken by $\overline{\mathsf{Fsk}}$ on all points in the class. As before, given any two $w_1, w_2 \in \mathcal{M}$, consider a permutation $\pi_{2\to1} \in \mathcal{P}$ that maps $w_2 \mapsto w_1$. Denote by $\mathcal{C}_1, \mathcal{C}_2$ the classes to which w_1, w_2 respectively belong. Using a similar argument as before, we see that the permutation $\pi_{2\to1}$ maps all points in $\mathcal{C}_2$ to points $\mathcal{C}_1$; since the converse holds for $(\pi_{2\to1})^{-1}$ we have that $\#\mathcal{C}_1 = \#\mathcal{C}_2$. The same argument applies for all pairs of classes.

Since the classes are disjoint we deduce that $\#\mathcal{C}_1 = ... = \#\mathcal{C}_n = \#\mathcal{M}/n \geq \#\mathcal{M}\,\#\mathcal{Q}/\#\mathcal{P}$. Consequently, regardless of the value of $w \in \mathcal{M}$, the information content of $\overline{\mathsf{Fsk}}[w]$ expressed in bits is (at most) $\leq \log_2[\#\mathcal{M}] - \log_2[\#\mathcal{M}\,\#\mathcal{Q}/\#\mathcal{P}] = \log_2[\#\mathcal{P}] - \log_2[\#\mathcal{Q}]$. In particular, for any random variable $W \in \mathcal{M}$ we have $\mathbf{H}_\infty[W] - \tilde{\mathbf{H}}_\infty[W \mid \overline{\mathsf{Fsk}}[W]] \leq \log_2[\#\mathcal{P}] - \log_2[\#\mathcal{Q}]$. $\square$

Proof of Claim 9.5. We are given a perturbation $\delta \in \Delta \subseteq \mathcal{P}$ and a collection of permutations $\overline{\mathsf{Fsk}}[w^*] \subseteq \mathcal{P}$. For all $\pi_p \in \overline{\mathsf{Fsk}}[w^*]$, define $\pi'_p = \pi_p \circ \delta^{-1} \in \mathcal{P}$. Consider the set $\{\pi'_p\} \subseteq \mathcal{P}$ of permutations obtained this way. We know that each $\pi'_p = \pi_p \circ \delta^{-1} = \pi_{p_2} \circ (\pi_{p_1} \circ \delta^{-1}) = \pi_{p_2} \circ \pi'_{p_1}$ where $\pi'_{p_1}[\delta[w^*]] = \omega_0$ and $\pi_{p_2} \subseteq \mathcal{Q}$. It follows that $\{\pi'_p\} \subseteq \overline{\mathsf{Fsk}}[\delta[w^*]]$. Similarly, for all $\pi' \in \overline{\mathsf{Fsk}}[\delta[w^*]]$ we know that $(\pi' \circ \delta) \in \overline{\mathsf{Fsk}}[w^*]$. It follows that $\{\pi'_p\}$ contains all of $\overline{\mathsf{Fsk}}[\delta[w^*]]$. Therefore $\overline{\mathsf{Fsk}}[\delta[w^*]]$ is precisely the set $\{\pi'_p\}$, whose elements are computable from available information. $\square$

A.2 Proving Theorem 11 (Generic Tightness)

Proof of Claim 11.1. The function $\overline{\overline{\mathsf{Fsk}}}$ partitions $\mathcal{M}$ into some number n of equivalence classes $\mathcal{C}_1, ..., \mathcal{C}_n$ such that $\forall w \in \mathcal{C}_i, \forall w' \in \mathcal{C}_j,\ i = j \Leftrightarrow \overline{\overline{\mathsf{Fsk}}}[w] = \overline{\overline{\mathsf{Fsk}}}[w']$. The equivalence classes can be delineated since $\overline{\overline{\mathsf{Fsk}}}$ is computable from $\langle \mathsf{Fsk}, \mathsf{Cor} \rangle$ and all the sets are finite. Since by construction $\overline{\overline{\mathsf{Fsk}}}$ is computable with arbitrarily high probability from the description of Cor and a randomized $\mathsf{Fsk} \circ \delta$ oracle for all $\delta \in \Delta$, it follows that the value of $\overline{\overline{\mathsf{Fsk}}}[w^*]$ can be determined from sufficiently many queries to $(\mathsf{Fsk} \circ \delta)[w^*]$, and from there which equivalence class the secret w^* belongs to. $\square$

Proof of Claim 11.2. The results follows immediately, for given the value of $\overline{\overline{\mathsf{Fsk}}}[w^*]$ it is easy to construct a perfect simulator for the randomized oracle $\mathsf{Fsk}[\delta[w^*]]$ on chosen input $\delta \in \Delta$. Consequently, $\overline{\overline{\mathsf{Fsk}}}[w^*]$ captures the total information that an unbounded adversary can extract from the challenger in a chosen perturbation outsider attack. $\square$

Proof of Claim 11.3. We bound the number n of classes as follows. Consider any random variable $W \in \mathcal{M}$ such that $\mathbf{H}_\infty[W] \geq m$. Since $\langle \mathsf{Fsk}, \mathsf{Cor} \rangle$ is a $(\mathcal{M}, m, m', t)$-fuzzy sketch, we know that $\bar{\mathbf{H}}_\infty[W \mid \mathsf{Fsk}[W]] \geq m'$. Since $\langle \mathsf{Fsk}, \mathsf{Cor} \rangle$ is unconditionally secure against outsider attacks, we also have $\bar{\mathbf{H}}_\infty[W \mid \overline{\overline{\mathsf{Fsk}}}[W]] \geq m'$. We construct a special random variable W as follows: first select a set of 2^m elements from $\mathcal{M}$ evenly spread out over all equivalence classes (*viz.*, $\lfloor 2^m/n \rfloor$ or $\lceil 2^m/n \rceil$ in each class); then define W to be the uniform distribution over this set. Clearly, $\mathbf{H}_\infty[W] = m$, and thus, $\bar{\mathbf{H}}_\infty[W \mid \overline{\overline{\mathsf{Fsk}}}[W]] \geq m'$. It follows that $\#\{\overline{\overline{\mathsf{Fsk}}}[w] : w \overset{\$}{\leftarrow} W\} \leq 2^{m-m'}$, from which we deduce that $n \leq 2^{m-m'}$ (unless W takes no more than one value in each class, which would imply $m' = 0$, in which case the claim would follow vacuously). $\square$

Proof of Claim 11.4. The equivalence structure that $\overline{\overline{\mathsf{Fsk}}}$ induces on $\mathcal{M}$ is preserved by Δ, *i.e.*, the images of all the points in any one class $\mathcal{C}_i$ under any perturbation $\delta \in \Delta$ all belong to a single class $\mathcal{C}_j$; otherwise we would have a procedure for gaining additional information about w^*, contradicting Claim 11.2. We write this as $\delta : \mathcal{C}_i \mapsto \mathcal{C}_j$. Since $\Delta' \subseteq \Delta$ is a generator set of the permutation group $\mathcal{P}$, it follows that the equivalence structure is preserved by $\mathcal{P}$. Furthermore, since $\mathcal{P}$ is a transitive group we have that for any pair of classes $\mathcal{C}_i, \mathcal{C}_j$ there exists $\pi \in \mathcal{P}$ such that $\pi : \mathcal{C}_i \mapsto \mathcal{C}_j$. It follows that all classes have the same cardinality, hence $\#\mathcal{C}_i = \#\mathcal{M}/n$. It also follows that $\mathcal{P}$ has a subset $\mathcal{Q}$ of size $\#\mathcal{Q} = \#\mathcal{P}/n$ that contains exactly the permutations that map the classes onto themselves, *i.e.*, for all $\pi \in \mathcal{Q}$ and all $\mathcal{C}_i$ we have $\pi : \mathcal{C}_i \mapsto \mathcal{C}_i$. It is easy to see that $\mathcal{Q}$ is a subgroup of $\mathcal{P}$, being closed under function inversion and function composition. Furthermore, $\log_2[\#\mathcal{P}] - \log_2[\#\mathcal{Q}] = \log_2[n] \leq m - m'$. $\square$

Proof of Claim 11.5. We start by refining our characterization of the information available to an unbounded adversary in an outsider attack, starting from $\overline{\mathsf{Fsk}}$ and $\overline{\overline{\mathsf{Fsk}}}$ as previously defined.

1. We define the function $\overline{\mathsf{Cor}} : \langle w', \textsc{p} \rangle \mapsto \{\mathsf{Cor}[w', \textsc{p}; r] : \forall r\}$ that maps any element w' and sketch $\textsc{p}$ to the set of all possible randomized values of $\mathsf{Cor}[w', \textsc{p}; r]$. Since the randomization set is finite, this function is computable from Cor, or with arbitrarily high probability given a randomized Cor oracle. Remark that $\overline{\mathsf{Cor}}$ essentially reduces to Cor when Cor is deterministic (as in the construction given in Section 6.2).

2. We define the function $\overline{\overline{\mathsf{Cor}}} : \textsc{p} \mapsto \{\langle w', \overline{\mathsf{Cor}}[w', \textsc{p}] \rangle : \forall w' \in \mathcal{M}\}$ that maps any sketch $\textsc{p}$ to the relation between elements $w' \in \mathcal{M}$ and all their randomized images $\mathsf{Cor}[w', \textsc{p}] \in \mathcal{M}$ (given as a set by $\overline{\mathsf{Cor}}[w', \textsc{p}] \subseteq \mathcal{M}$). Since $\mathcal{M}$ is finite, this function is computable from $\overline{\mathsf{Cor}}$.

3. We define the function $\overline{\overline{\overline{\mathsf{Cor}}}} : w \mapsto \{\langle \textsc{p}, \overline{\overline{\mathsf{Cor}}}[\textsc{p}] \rangle : \forall \textsc{p} \in \overline{\mathsf{Fsk}}[w]\}$ that maps any element $w \in \mathcal{M}$ to the relation between the values assumed by Fsk on input w (given by $\overline{\mathsf{Fsk}}[w]$) and the corresponding values of $\overline{\overline{\mathsf{Cor}}}$. This function is computable from $\overline{\overline{\mathsf{Cor}}}$ for any value of $\overline{\mathsf{Fsk}}[w]$.

4. We define the function $\overline{\overline{\overline{\mathsf{Fsk}}}} : w \mapsto \{\langle \delta, \overline{\overline{\overline{\mathsf{Cor}}}}[\delta[w]] \rangle : \forall \delta \in \Delta\}$ that maps any element $w \in \mathcal{M}$ to the relation between perturbations $\delta \in \Delta$ and the values taken by $\overline{\overline{\overline{\mathsf{Cor}}}}$ on input $\delta[w]$. Since the set Δ is finite, this function can be computed from $\overline{\mathsf{Fsk}}$ and $\overline{\overline{\overline{\mathsf{Cor}}}}$.

$\overline{\overline{\overline{\mathsf{Fsk}}}}$ and $\overline{\overline{\mathsf{Fsk}}}$ are the same in terms of information content; indeed, $\overline{\overline{\overline{\mathsf{Fsk}}}}$ is an annotated version of $\overline{\overline{\mathsf{Fsk}}}$ that indicates the behavior of Cor on all possible inputs under all possible sketches for all admissible perturbations of the (unknown) secret w^*. Since all these annotations are computed, rather than queried, by the adversary, they provide no additional information. In particular, the same equivalence classes are induced on $\mathcal{M}$ by $\overline{\overline{\mathsf{Fsk}}}$ and $\overline{\overline{\overline{\mathsf{Fsk}}}}$.

Now, to prove our claim, we show that each equivalence class by itself forms a $(\mathcal{M}, K, t)$-code of size $K = \#\mathcal{M}/n$. To see this, fix one such class $\mathcal{C}$. Take some arbitrary $w_0 \in \mathcal{C}$ and r_0, and let $\mathrm{P}_0 \leftarrow \mathsf{Fsk}[w_0; r_0]$; notice that $\mathsf{Cor}[w_0, \mathrm{P}_0] = w_0$. Since Δ contains the identity perturbation δ_0, we know that $\overline{\overline{\overline{\mathsf{Fsk}}}}[w_0]$ contains a tuple

$$\begin{aligned} &\langle\ \delta_0\ ,\ \{\langle \mathsf{Fsk}[\delta_0[w_0]; r_0], \{\langle w_0, \{\mathsf{Cor}[w_0, \mathrm{P}_0]\}\rangle, ...\}\rangle, ...\}\ \rangle \\ =\ &\langle\ \delta_0\ ,\ \{\langle \mathrm{P}_0, \{\langle w_0, \{w_0\}\rangle, ...\}\rangle, ...\}\ \rangle \qquad\qquad \in\ \overline{\overline{\overline{\mathsf{Fsk}}}}[w_0]\ , \end{aligned}$$

which simply captures the knowledge that on input $\delta_0[w_0] = w_0$ there is a random execution of the sketching function Fsk that returns P_0, and that given this fuzzy sketch P_0 the correction function Cor maps the word w_0 to itself (deterministically in this case). Since $\overline{\overline{\overline{\mathsf{Fsk}}}}[w_0]$ is invariant with respect to the particular choice of w_0 within $\mathcal{C}$, it follows that for all $w \in \mathcal{C}$ the same tuple $\langle \delta_0, \{\mathrm{P}_0, ...\}, \{\langle \mathrm{P}_0, \{\langle w_0, w_0\rangle, ...\}\rangle, ...\}\rangle$ also appears in $\overline{\overline{\overline{\mathsf{Fsk}}}}[w]$. This requires that there be a random execution of Fsk that on input $\delta_0[w] = w$ outputs P_0. Now, since the fuzzy sketch is assumed able to correct all errors within a radius of t, it follows that for all neighboring $w' \in \mathcal{M}$ such that $\mathsf{d}[w, w'] \leq t$ we must have $\mathsf{Cor}[w', \mathrm{P}_0] = w$. Therefore, generalizing over all $w \in \mathcal{C}$ we see that the function $D : w' \mapsto \mathsf{Cor}[w', \mathrm{P}_0]$ is the decoding function of a $(\mathcal{M}, K, t)$-code whose codebook is precisely $\mathcal{C}$. Since all the permutations in $\mathcal{Q}$ map the classes onto themselves, any point $\omega_0 \in \mathcal{C}$ will serve as $\mathcal{Q}$-pivot of $\mathcal{C}$, as required. □

B Ideal Insider Security Proof

We sketch a proof of the main theorem of Section 7. For clarity, we first spell out the full construction, and motivate its principle.

Full Construction. As before, we let $\omega_0 \in \mathcal{C} \subseteq \mathcal{M}$ and $\mathcal{Q} \subseteq \mathcal{P}$ be as in Section 6, and define the fully randomized generic fuzzy sketch $\langle \mathsf{Fsk}, \mathsf{Cor}\rangle$ exactly as in Section 7:

$$\mathsf{Fsk}[w; r] = \mathrm{P} \quad \begin{cases} p_1 \stackrel{r}{\leftarrow} \{p' : \pi_{p'} \in \mathcal{P}, \pi_{p'}[w] = \omega_0\} \\ p_2 \stackrel{r}{\leftarrow} \{p'' : \pi_{p''} \in \mathcal{Q}\} \\ \mathrm{P}\ \text{ s.t. } \pi_{\mathrm{P}} = \pi_{p_2} \circ \pi_{p_1} \in \mathcal{P} \end{cases} \qquad \begin{aligned} \mathsf{Cor}[w', \mathrm{P}] &= (\pi_{\mathrm{P}}^{-1} \circ \pi^{-1} \circ C \circ D \circ \pi \circ \pi_{\mathrm{P}})[w'] \\ &\text{for random } \ \pi \leftarrow \mathcal{Q}\ . \end{aligned}$$

Next, assuming a hash function H which will be viewed as a random oracle, we define the full generic fuzzy extractor $\langle \mathsf{Gen}, \mathsf{Reg}\rangle$ as follows:

$$\mathsf{Gen}[w; \langle r, r'\rangle] = \langle \mathrm{S}, \mathrm{Q}\rangle \quad \text{where} \quad \begin{cases} \mathrm{P} = \mathsf{Fsk}[w; r] \\ \mathrm{Q} = \langle \mathrm{P}, r'\rangle \\ \mathrm{S} = \mathsf{H}[w, r', \mathrm{P}] \end{cases} \qquad \begin{aligned} \mathsf{Reg}[w', \mathrm{Q}] &= \mathsf{H}[\mathsf{Cor}[w', \mathrm{P}], r', \mathrm{P}] \\ &\text{where } \ \langle \mathrm{P}, r'\rangle = \mathrm{Q}\ . \end{aligned}$$

Here, H is viewed as a random function from $\mathcal{M} \times \{0,1\}^{\ell'} \times \{0,1\}^{\ell''}$ into $\{0,1\}^{\ell}$, where we assume that the sketch strings P can be represented in $\{0,1\}^{\ell''}$. The second input to H is an explicit ℓ'-bit randomization argument, which also allows us to treat H as an optimal randomness extractor.

Randomness Extraction. For any fixed (but secret) $\textsc{p}$ we can view the map $\mathsf{Ext}_{\textsc{p}} \equiv \mathsf{H}[\cdot, \cdot, \textsc{p}]$ as an optimal (n, m', ℓ, ϵ)-randomness extractor in the sense of Section 3. Indeed, since $\mathsf{Ext}_{\textsc{p}}$ is a random function, it is an optimal randomness extractor in the sense of Section 3.1, and, thus, for any m-bit min-entropy distribution W and uniform R, satisfies $\mathbf{D}[\langle \mathsf{Ext}_{\textsc{p}}[W, R], R\rangle, \langle U_\ell, R\rangle] \leq \epsilon$ where $\epsilon = \sqrt{2^{\ell-m}}$ according to the bound given in [RTS97]. In the case at hand, the value of $\textsc{p}$ is public, and drawn from the distribution $\mathsf{Fsk}[W]$. Since we have assumed that $\bar{\mathbf{H}}_\infty[W \mid \textsc{p}] \geq m'$, we obtain by the same argument that $\mathbf{D}[\langle \mathsf{Ext}_{\textsc{p}}[W, R], R, \textsc{p}\rangle, \langle U_\ell, R, \textsc{p}\rangle] = \mathbf{D}[\langle \mathsf{H}[W, R, \textsc{p}], R, \textsc{p}\rangle, \langle U_\ell, R, \textsc{p}\rangle] \leq \epsilon$ where $\epsilon = \sqrt{2^{\ell-m'}}$. We use this bound in the sequel of the argument.

A Hall Of Mirrors. Recall that the security of the fuzzy sketch construction of Section 6.2 was based on the weak notion of symmetric subcode, which we used to keep the outsider adversary in the dark. We designed Fsk to ensure that all responses to public queries would "map" into the symmetric subcode $\mathcal{C}' \subseteq \mathcal{C}$, steering clear from any potentially recognizable "landmark" lurking in $\mathcal{C} \setminus \mathcal{C}'$.

When private queries are added, the situation is more delicate, since with the original definition of Cor an appropriately chosen query $\langle \delta, \textsc{q}\rangle$ can force the challenger to "correct" any secret w^* to any codeword in $\mathcal{C}$, not just $\mathcal{C}'$. We cannot prevent this from happening, but we can randomly shuffle things around to prevent such occurences to leak any additional information about w^*.

Specifically, the "code action" $(C \circ D)$ is replaced by $(\pi^{-1} \circ (C \circ D) \circ \pi)$ for a randomly chosen permutation $\pi \in_\$ \mathcal{Q}$. This causes Cor to behave as if it were using a different code $\mathcal{C}_\pi = \pi[\mathcal{C}]$, randomly instantiated upon each invocation as the code image of $\mathcal{C}$ under a fresh random permutation $\pi \in_\$ \mathcal{Q}$. Since the subcode $\mathcal{C}'$ is invariant under $\mathcal{Q}$, the behavior of Cor is unchanged (and thus deterministic) on all input w' within distance t of any codeword in $\mathcal{C}'$; hence, the required correction properties are preserved. The point of the random shuffling it to render all codewords indistinguishable up to permutations in $\mathcal{Q}$.

Random Blinding. We now turn to Reg to assess the knowledge gained by the adversary from private queries.

Since H is a random oracle, the only useful knowledge that the adversary can gain from private queries is which of its outputs are equal, helping it to deduce equality relationships between the function indices. However, since the random oracle takes $\textsc{p}$ and r' as part of its input, no useful information will be gained between any two private queries that involve distinct values of $\textsc{p}$ or r'.

Hence, the only knowledge that the adversary gains from a collection of private queries $\langle \delta_k, \textsc{q}_k\rangle$ for $k = 1, ..., q'$ is contained in a set of (positive or negative) pairwise equality relationships δ_{ij} such that $\delta_{ij} = 1$ if $\langle \textsc{p}_i, r'_i, w^*_i\rangle = \langle \textsc{p}_j, r'_j, w^*_j\rangle$ and $\delta_{ij} = 0$ otherwise, posing $w^*_k \leftarrow \mathsf{Cor}[\delta_k[w^*], \textsc{p}_k]$ in the k-th private query (where naturally the values of w^*_k remain hidden from the adversary's view).

Knowledge Limitation. Consider the following "partial" randomized correction function:

$$\mathsf{pCor}[w', \textsc{p}; r] = (C \circ D \circ \pi \circ \pi_{\textsc{p}})[w'] \qquad \text{where} \;\; \pi \xleftarrow{r} \mathcal{Q} \; .$$

Define $\overline{\mathsf{pCor}} : \langle w', \textsc{p}\rangle \mapsto \{\mathsf{pCor}[w', \textsc{p}; r] : \forall r\}$ to characterize all the knowledge one can obtain by sampling a pCor oracle. Clearly, the information about w^* contained in the $\binom{q'}{2}$ pairwise equality relationships δ_{ij} for $i, j = 1, ..., q'$ is subsumed by the information contained in the tuples $\langle \textsc{p}_k, r'_k, \overline{\mathsf{pCor}}[\delta_k[w^*], \textsc{p}_k]\rangle$ for $k = 1, ..., q'$. Indeed, given these tuples the δ_{ij} can be simulated by sampling $w_i \xleftarrow{\$} \overline{\mathsf{pCor}}[\delta_i[w^*], \mathrm{P}_i]$ and $w_j \xleftarrow{\$} \overline{\mathsf{pCor}}[\delta_j[w^*], \mathrm{P}_j]$, comparing $w_i \stackrel{?}{=} w_j$, and assigning $\delta_{ij} \leftarrow 1$ only in case of equality (note that this is sound only for matching $\mathrm{P}_i = \mathrm{P}_j$, but fortunately δ_{ij} is always zero failing this condition).

Now, recall from the proof of Theorem 9 the definition of $\overline{\mathsf{Fsk}}$ and the equivalence classes it defines on $\mathcal{M}$. It can be shown using a technique similar to that proof that $\overline{\mathsf{Fsk}}$ and $\overline{\mathsf{pCor}}$ define the same equivalence classes over $\mathcal{M}$. Consequently, the insider adversary obtains no information from private queries that is not already available to an unbounded outsider adversary making public queries only, unless there was a collision.

Direct Guesses. Notwithstanding any of the above, the adversary can use private queries to guess and test the challenger's secret, one candidate value at a time. Over q' private queries this attack will produce a correct guess with probability no greater than $q'\, 2^{-m'}$.

In addition, the adversary can guess the target private string (rather than the challenger's secret) when it produces its final answer. Since the secret $w^* \xleftarrow{\$} W$ where $\mathbf{H}_\infty[W \mid \mathsf{Fsk}[W]] \geq m'$ we know that the randomness extractor $\mathsf{H}[w^*, \ldots]$ deviates from the uniform distribution with a statistical error no greater than ϵ, so the probability of the adversary's making a successful final blind guess is also bounded by ϵ.

Hash Collisions. In the presence of non-trivial collisions, all bets are off. However, since H is a random oracle with ℓ bits of output, the pairwise probability of collision under H is at most $2^{-\ell}$. Over q' private queries (where q' also includes the number of direct queries to the random oracle), the probability of encountering at least one non-trivial collision is thus no greater than $\binom{q'}{2}\, 2^{-\ell}$.

Putting it all together, we are now in a position to prove the following, restated version of Theorem 12:

Theorem 13. *Under the same conditions as in Theorem 9 where the code $\mathcal{C}$ has error correction limit $\leq \bar{t}$, the above algorithms $\langle \mathsf{Gen}, \mathsf{Reg} \rangle$ constitute a $(\mathcal{M}, m, \ell, t, \epsilon)$-fuzzy extractor that is $(\infty, \infty, q', \bar{t}, \alpha)$-OW-Fuz-CPA and IND-Fuz-CPA secure whenever $\alpha \geq \binom{q'}{2}\, 2^{-\ell} + q'\, 2^{-m'} + \epsilon$, in the random oracle model, where q' also includes the number of direct queries to the random oracle.*

Proof. The proof is organized along successive claims. First, we know the following.

Claim 13.1. The above construction gives a $(\mathcal{M}, m, \ell, t, \epsilon)$-fuzzy extractor. In particular, Cor is deterministic on all input words w' within distance t of any codeword in $\mathcal{C}'$, and has the required correction properties on such inputs.

Claim 13.2. The fuzzy extractor is unconditionally secure against outsider attacks.

Next, we show that, if all goes well, the information gained from q' private queries reduces to a set of positive and negative equality relations, thanks to the blinding effect of random hashing.

Claim 13.3. In the absence of non-trivial collisions, *i.e.*, private queries $\langle \mathrm{Q}_i, \delta_i \rangle$ and $\langle \mathrm{Q}_j, \delta_j \rangle$ where $\mathrm{Q}_i = \mathrm{Q}_j = \langle \mathrm{P}, \ldots \rangle = \mathrm{Q}$ such that $\mathsf{Cor}[\delta_i[w^*], \mathrm{P}] \neq \mathsf{Cor}[\delta_j[w^*], \mathrm{P}]$ but $\mathsf{Reg}[\delta_i[w^*], \mathrm{Q}] = \mathsf{Reg}[\delta_j[w^*], \mathrm{Q}]$, the information gained by the adversary from a collection of private queries $\langle \delta_k, \mathrm{Q}_k \rangle$ for $k = 1, \ldots, q'$, is that of pairwise equality relations "δ_{ij}" between the tuples $\langle \mathrm{P}_k, r'_k, w^*_k \rangle$ where $w^*_k \xleftarrow{\$} \mathsf{Cor}[\delta_k[w^*], \mathrm{P}_k]$ is obtained by sampling.

Then, we show that these equality relations themselves contain no more information about w^* than is already available from (unbounded) public queries, thanks to the randomization of Cor.

Claim 13.4. Barring non-trivial collisions, the above set of relations δ_{ij} contains no more information than the set of tuples $\langle \mathrm{P}_k, r'_k, \overline{\mathsf{pCor}}[\delta_k[w^*], \mathrm{P}_k]\rangle$ for $k = 1, ..., q'$, where $\overline{\mathsf{pCor}}$ is defined as above.

Claim 13.5. The values of $\overline{\mathsf{pCor}}[\delta_k[w], \mathrm{P}_k]$ on queries $\langle \mathrm{Q}_k, \delta_k\rangle$ remain invariant when w varies within any fixed equivalence class $\mathcal{C}_i \subseteq \mathcal{M}$ as induced by $\overline{\mathsf{Fsk}}$, where $\overline{\mathsf{Fsk}}$ is defined as is Section 6.3.

Further, we show that, due to random hashing and the error correction limit, the adversary cannot directly query for the target and can only obtain information about it through a non-trivial collision.

Claim 13.6. The adversary gains no information about the target private string from private queries not involving the challenge public string.

Claim 13.7. Absent non-trivial collisions, the adversary learns nothing about the target private string from private queries that involve the challenge string while abiding by the minimum displacement requirement.

Last, we bound the adversary's chance of merely guessing the hidden secret outright during the course of the attack, during both the query phase and the challenge phase, and finally correct for the probability of occurrence of a non-trivial collision in the sense of Claim 13.3.

Claim 13.8. Barring non-trivial collisions, the adversary can use each private query to guess and test one candidate for the challenger's secret with success probability no greater than $2^{-m'}$. The adversary can independently guess the target private string it outputs with success probability at most ϵ. The total probability of the adversary's making at least one such successful direct guess is thus $\leq q'\, 2^{-m'} + \epsilon$.

Claim 13.9. For any $\mathrm{Q} = \langle \mathrm{P}, r'\rangle$ the collision probability of the random function $\mathsf{H}[\cdot, r', \mathrm{P}] = \lambda w.\mathsf{H}[w, r', \mathrm{P}]$ on an arbitrary pair of distinct inputs is $2^{-\ell}$. Hence, the probability of non-trivial collisions on any sequence of q' private queries is thus $\leq \binom{q'}{2}\, 2^{-\ell}$.

The theorem now follows easily from all the above claims. $\square$

The subclaims are easily seen to hold from the preceding discussion, with the exception of Claim 13.5, which requires more work similar to the proof given in Section A.2.

C Explicit Construction Of A Biased Code

To illustrate a point made in Section 4.2, we give a (very simple and suboptimal) construction of a biased binary code, *i.e.*, a binary code with the property that for uniformly distributed input to the encoding function, every coordinate of the output codeword is more likely to be 0 than 1.

Specifically, given a parameter $\gamma \geq 3$ and an input dimension k assumed to be a multiple of γ, we construct an efficiently decodable non-linear binary code of parameters $[n, k, d]$, where $n = k\,(2^\gamma - 1)/\gamma$ and $d = 2^{\gamma-1} - 1$.

We emphasize that the following construction is very suboptimal, in the sense that there are more efficient ways to obtain a biased code. However, it has the benefit of providing a keen intuition into the property of fuzzy sketches that we seek to exploit.

Tweaked Hadamard Matrices. Recall that a Hadamard matrix of order h is a $h \times h$ matrix with elements in $\{0, 1\}$ such that the Hamming distance between any two distinct rows or columns is exactly $h/2$. For any power-of-two order $h = 2^\gamma$, it is easy to construct a Hadamard matrix $\mathcal{H}_h$ with all 1's on the first column and on the last row. Here are the first few such matrices (writing $0, 1$ as $-, +$ for readability):

$$\mathcal{H}_1 = \begin{pmatrix} + \end{pmatrix} , \quad \mathcal{H}_2 = \begin{pmatrix} + & - \\ + & + \end{pmatrix} , \quad \mathcal{H}_4 = \begin{pmatrix} + & - & - & + \\ + & + & - & - \\ + & - & + & - \\ + & + & + & + \end{pmatrix} , \quad \mathcal{H}_8 = \begin{pmatrix} + & - & - & + & - & + & + & - \\ + & + & - & - & - & - & + & + \\ + & - & + & - & - & + & - & + \\ + & + & + & + & - & - & - & - \\ + & - & - & + & + & - & - & + \\ + & + & - & - & + & + & - & - \\ + & - & + & - & + & - & + & - \\ + & + & + & + & + & + & + & + \end{pmatrix} ,$$

$$\mathcal{H}_{16} = \begin{pmatrix} + & - & - & + & - & + & + & - & - & + & + & - & + & - & - & + \\ + & + & - & - & - & - & + & + & - & - & + & + & + & + & - & - \\ + & - & + & - & - & + & - & + & - & + & - & + & + & - & + & - \\ + & + & + & + & - & - & - & - & - & - & - & - & + & + & + & + \\ + & - & - & + & + & - & - & + & - & + & + & - & - & + & + & - \\ + & + & - & - & + & + & - & - & - & - & + & + & - & - & + & + \\ + & - & + & - & + & - & + & - & - & + & - & + & - & + & - & + \\ + & + & + & + & + & + & + & + & - & - & - & - & - & - & - & - \\ + & - & - & + & - & + & + & - & + & - & - & + & - & + & + & - \\ + & + & - & - & - & - & + & + & + & + & - & - & - & - & + & + \\ + & - & + & - & - & + & - & + & + & - & + & - & - & + & - & + \\ + & + & + & + & - & - & - & - & + & + & + & + & - & - & - & - \\ + & - & - & + & + & - & - & + & + & - & - & + & + & - & - & + \\ + & + & - & - & + & + & - & - & + & + & - & - & + & + & - & - \\ + & - & + & - & + & - & + & - & + & - & + & - & + & - & + & - \\ + & + & + & + & + & + & + & + & + & + & + & + & + & + & + & + \end{pmatrix} .$$

Every column but the first and every row but the last contain the same number of 0's and 1's. For each such matrix $\mathcal{H}_h$ with h a power of two, we define the modified matrix $\bar{\mathcal{H}}_h$ of size $h \times (h - 1)$ by removing the first column and replacing all 1's on the last row by 0's. We obtain:

$$\bar{\mathcal{H}}_1 = \emptyset , \quad \bar{\mathcal{H}}_2 = \begin{pmatrix} - \\ - \end{pmatrix} , \quad \bar{\mathcal{H}}_4 = \begin{pmatrix} - & - & + \\ + & - & - \\ - & + & - \\ - & - & - \end{pmatrix} , \quad \bar{\mathcal{H}}_8 = \begin{pmatrix} - & - & + & - & + & + & - \\ + & - & - & - & - & + & + \\ - & + & - & - & + & - & + \\ + & + & + & - & - & - & - \\ - & - & + & + & - & - & + \\ + & - & - & + & + & - & - \\ - & + & - & + & - & + & - \\ - & - & - & - & - & - & - \end{pmatrix} ,$$

$$\bar{\mathcal{H}}_{16} = \begin{pmatrix} - & - & + & - & + & + & - & - & + & + & - & + & - & - & + \\ + & - & - & - & - & + & + & - & - & + & + & + & + & - & - \\ - & + & - & - & + & - & + & - & + & - & + & + & - & + & - \\ + & + & + & - & - & - & - & - & - & - & - & + & + & + & + \\ - & - & + & + & - & - & + & - & + & + & - & - & + & + & - \\ + & - & - & + & + & - & - & - & - & + & + & - & - & + & + \\ - & + & - & + & - & + & - & - & + & - & + & - & + & - & + \\ + & + & + & + & + & + & + & - & - & - & - & - & - & - & - \\ - & - & + & - & + & + & - & + & - & - & + & - & + & + & - \\ + & - & - & - & - & + & + & + & + & - & - & - & - & + & + \\ - & + & - & - & + & - & + & + & - & + & - & - & + & - & + \\ + & + & + & - & - & - & - & + & + & + & + & - & - & - & - \\ - & - & + & + & - & - & + & + & - & - & + & + & - & - & + \\ + & - & - & + & + & - & - & + & + & - & - & + & + & - & - \\ - & + & - & + & - & + & - & + & - & + & - & + & - & + & - \\ - & - & - & - & - & - & - & - & - & - & - & - & - & - & - \end{pmatrix} .$$

We note the following facts regarding the modified matrices $\bar{\mathcal{H}}_h$:

1. the rows of $\bar{\mathcal{H}}_h$ have pairwise Hamming distance at least $h/2 - 1$;
2. each column of $\bar{\mathcal{H}}_h$ contains precisely two more 0's than 1's.

The Biased Code. The code $C : \{0, 1\}^k \to \{0, 1\}^n$ is constructed as follows: on input a bit vector $r \in \{0, 1\}^k$, segment r into chunks of γ bits each; then, interpreting each chunk as a value in $\{1, ..., 2^\gamma\}$, replace that chunk by the corresponding row in the matrix $\bar{\mathcal{H}}_{2^h}$. The result is a bit vector of length $n = (k/\gamma)\,(2^\gamma - 1)$, which is retained as the output of $C[r]$. The code admits an efficient decoding function $D : \{0, 1\}^n \to \{0, 1\}^k$ that works by segmenting its input into chunks of size $(2^\gamma - 1)$, matching each chunk at a time to the closest row of $\bar{\mathcal{H}}_{2^h}$, and transcribing the resulting row indices into as many concatenated γ-bit numbers.

Embracing JW-DRS. When the JW-DRS construction of Section 3.3 is applied to the code C, we obtain a $(\mathcal{M}, m, \ell, t, \epsilon)$-fuzzy extractor $\langle \mathsf{Gen}, \mathsf{Reg} \rangle$ where $\mathcal{M} = \{0, 1\}^n$ for $n = k\,(2^\gamma - 1)/\gamma$, $\ell = m + k - n - 2\,\log_2[1/\epsilon]$, and $t = 2^{\gamma - 2} - 1$. Since Gen takes a noisy input of size n, it is legitimate to require that the input distribution W have min-entropy $\mathbf{H}_\infty[W] \geq m$ as long as $m \leq n$. Thus, for any choice of $\gamma \geq 3$ such that $\gamma \mid k$, there is a feasible set of values for m and ϵ for which the JW-DRS construction gives a non-trivial fuzzy extractor based on the code C.

Defeating JW-DRS. From our previous observations about the matrices $\bar{\mathcal{H}}$, it follows that for uniformly distributed randomness $r \in_{\$} \{0,1\}^k$ the codewords $C[r] \in \{0,1\}^n$ will be such that at each coordinate the value 0 will appear more often than the value 1. More precisely, for each coordinate $j = 1, ..., n$, we have:

$$\mathbf{P}[C[r]|_j = 0] \geq \frac{1}{2} + \frac{1}{h} = \frac{1}{2} + 2^{-\gamma} \ .$$

Since the public string $\mathtt{Q}$ returned by Gen on input w^* contains the substring $w^* \oplus C[r]$ for some r chosen uniformly at random, it follows that it is easy to recover all the bits of w^* given a sufficiently large number $q = 2^{\Theta[\gamma]} = \mathcal{O}[\text{poly}[t]]$ of these public strings (provided that they are computed independently from the same input word w^*): simply do a majority vote over all q public strings, one coordinate at a time.

Noise Tolerance. We note that the above attack is robust to small Hamming perturbations of w^*. In other words, if instead of Gen being applied to the same secret w^* it is applied to small Hamming perturbations $w_1, ..., w_q$ thereof (within a ball of radius t centered on w^*), then the above attack will produce a word $\tilde{w}$ that is equally close to w^* (possibly closer if the various perturbations cancel on average). In virtue of the error tolerance properties of fuzzy extraction, using $\tilde{w}$ instead of w^* the attacker will still be able to exactly regenerate all the extracted private strings $\mathtt{S}_1, ..., \mathtt{S}_q$.