

Strongly Robust Fuzzy Extractors

Abstract

Fuzzy extractors are used to generate reliably reproducible randomness from a biased, noisy source. Known constructions of fuzzy extractors are built from a strong extractor, and a *secure sketch*, a function that transforms a biased noisy secret value into a public value, simultaneously hiding the secret and allowing for error correction. A *robust* sketch is secure against adversarial modification: no adversary can make a new valid sketch of a secret after seeing one valid sketch of that secret. Prior constructions of robust sketches are proven secure against an unbounded adversary that sees *one and only one* valid sketch of a secret.

In this paper we examine the notion of *strong robustness*, that is, robustness even when the adversary receives multiple sketches of related secrets. Strong robustness can be used to prove that a fuzzy extractor is secure in a fully adaptive setting (called “insider security” by Boyen [3]). We demonstrate that previous secure sketches are not strongly robust, and give a proof of impossibility which demonstrates that sketches cannot be strongly robust against an unbounded adversary, for any reasonable set of perturbations. We then give two constructions of sketches that are strongly robust against a computationally bounded adversary. The first construction is proven secure assuming the existence of an xor related-key secure MAC in the CRS model, while the second construction is proven in the random oracle model.

We show that our constructions can be adapted in the natural way into a strongly robust fuzzy extractor, and we demonstrate that these strongly robust fuzzy extractors are insider secure. It remains an open problem [3] to find a fuzzy extractor that is insider secure against an unbounded adversary, but our impossibility result implies that one cannot achieve such an extractor via robustness.

1 Introduction

Secure sketches and fuzzy extractors, first proposed by Dodis, Reyzin and Smith [10], allow cryptographic protocols to rely on fuzzy secrets that are neither uniformly random nor exactly reproducible. A secure “sketch” of a value keeps the value secret, yet allows for recovery from any “close” value. A fuzzy extractor allows for the reliable (re)generation of a random key from a noisy, biased, secret piece of information. All known fuzzy extractors are built by combining fuzzy sketches and standard strong extractors.

Building from this foundational work, Boyen, Dodis, Katz, Ostrovsky and Smith [4] introduced the idea of a *robust* sketch, a sketch where it is hard for an adversary to produce a new valid sketch of a secret, after seeing one, and (only) one valid sketch. All prior work on robust sketches has focused exclusively on this “one-time” notion.

In this paper, we consider the notions of *strongly robust* sketches and fuzzy extractors to be sketches/extractors that are robust even in the presence of multiple valid sketches. This is an important strengthening of the (one-time) notion of robustness for several reasons. First, it is the most natural notion for robustness, and mirrors similar definitions for MACs and signature schemes. Second, several prior schemes are easily attackable in this stronger model, which we demonstrate in Appendix C. Third, many applications of fuzzy extractors either make artificially strong assumptions or admit the use of multiple sketches (see Appendix B for analysis). Fourth, the strongest notion of security for fuzzy extractors, the “insider security” notion of Boyen [3], can be easily achieved through a strongly robust fuzzy extractor, and it is an open problem to find an insider-secure fuzzy extractor without the random oracle model.

1.1 Our results

In this paper, we first formally define the notion of strong robustness. We then give an impossibility result to show that no sketch can be strongly robust against an unbounded adversary, except with long shared keys, for any reasonable class of perturbations. This implies that no prior robust sketch is strongly robust. See Appendix C for a discussion of more efficient attacks on prior robust fuzzy extractor schemes. We then give two keyless constructions of a fuzzy extractor which is strongly post-application robust against computationally bounded adversaries, by adapting the construction of Cramer et al. [7]. Our first construction is based on the assumption that xor related-key secure MACs exist, and is proven in the CRS model. Our second construction is in a query-bounded random oracle model, and can be viewed as an extension of Boyen’s construction [3] to provide robustness. We finally show how our construction can be extended to an insider secure fuzzy extractor, as long as our reusable sketch has a rather standard “linearity” property.

1.2 Prior work

Many methods to achieve secret key agreement from noisy or biased information (such as biometrics) in an authenticated channel have been shown [2, 1, 12, 11, 14, 9].

Secure sketches and fuzzy extractors were first proposed in 2004 by Dodis, Reyzin and Smith [10] as a methodology for turning noisy information into secure random keys over an unauthenticated channel. Several different constructions of secure sketches have been given [10, 6, 5], for varying choices of the underlying distance metric. Boyen showed that prior definitions are not adequate to cases in which the fuzzy secret is used multiple times, and defines the notion of a reusable sketch which addresses this problem [3]. In the same paper, Boyen defined the notion of an insider-secure fuzzy extractor, which is secure in the presence of multiple sketches and “insider queries” that allow an adversary to learn extracted keys.

When a fuzzy extractor is used for the purposes of authentication, there remains the possibility of an adversary modifying the sketch as it is sent across the communications channel, which could lead to a form of man-in-the-middle attack. To avoid this, Boyen et al. defined the notion of a *robust* sketch: a sketch for which no adversary can produce a valid new sketch after seeing one [4]. Boyen et al. made a keyless¹, statistically robust sketch in the random oracle model. Several subsequent improvements have been described in the literature using the same basic “one-time robustness” definition. Dodis et al. constructed a keyless, statistically robust sketch in the plain model [8]. Cramer et al. [7] and Kanukurthi and Reyzin [13] give robust sketches that lead to fuzzy extractors that produce relatively longer outputs for similar parameters; the former in the common random string model, the latter in the plain model. So far we know, our results are the first on *multi-use* robust sketches.

2 Preliminaries

In this section, we introduce notation and overview definitions that we use throughout the paper. For a randomized algorithm F , we denote $F(a; r)$ as F running on input a with randomness r . We denote a random variable over a set \mathcal{W} as W . We denote $w \leftarrow W$ as an element of \mathcal{W} sampled

¹Note that if the sender and recipient of a secure sketch share a key, this would imply an authenticated channel. So, only keyless constructions, or constructions with very short keys, are of any interest.

according to W . We denote $w \leftarrow W$ as w uniformly selected from W . If a set \mathcal{M} is a metric space then we denote the distance function on \mathcal{M} as $\|x - x'\|$ for $x, x' \in \mathcal{M}$.

Definition 2.1 (Statistical difference) *The statistical difference of two random variables W, W' over a common domain W as is defined as $SD(W, W') = \frac{1}{2} \sum_{w \in W} |Pr[w \leftarrow W] - Pr[w \leftarrow W']|$.*

Definition 2.2 (Min-entropy) *The min-entropy of a random variable W is defined as:*

$$H_\infty(W) = -\log(\max_a Pr[a \leftarrow W])$$

Definition 2.3 (Conditional min-entropy) *The average conditional min-entropy of a variable W given W' is*

$$H_\infty(W|W') = -\log(\mathbb{E}_{b \leftarrow W'}[2^{-H_\infty(W|W'=b)}])$$

where \mathbb{E} denotes expectation.

Definition 2.4 ((d, m) -pair) *Two distributions W, W' over \mathcal{M} are called a (d, m) -pair if they have the property that the distance between any two points $w \in W, w' \in W'$ is $\leq d$ and $H_\infty(W) \geq m$.*

Definition 2.5 *We say that a f is negligible in n if, for all constants $c > 0$ there exists n_0 such that $\forall n > n_0, f(n) \leq \frac{1}{n^c}$.*

2.1 Codes

An *error-correcting code* is a code $\mathcal{C} \subset \mathcal{M}$ along with a triple of algorithms (C, C^{-1}, D) where: [3, 9]

1. C is the encoding function which takes elements of a domain of size 2^k to \mathcal{C}
2. C^{-1} is the inverse of C
3. $\exists t : \forall m \in \mathcal{C}, \forall m' : \|m' - m\| \leq t, D(m') = m$.

We refer to the tuple $(\mathcal{C}, C, C^{-1}, D)$ as an n, k, t error correcting code.

A $[n, k, d]$ linear code is a code where \mathcal{C} is a k -dimensional linear subspace of \mathcal{M} of dimension n , and where \mathcal{C} has minimum distance d . As such $C(x) = \mathbf{C}x$ for a generator matrix \mathbf{C} . We note also that all linear codes have a parity check matrix, a matrix \mathbf{H} where $\mathbf{H}C(x) = 0$.

An extractor uses a short, truly random seed and a longer biased source of randomness, to produce an output statistically close to uniform.

Definition 2.6 *An (n, m', l, ϵ) -strong extractor is a randomized algorithm $\text{Ext}\{0, 1\}^n \rightarrow \{0, 1\}^l$ such that for any random variable W with min-entropy m' we have that $SD(\langle \text{Ext}(W; R), R \rangle, \langle U_l, R \rangle) \leq \epsilon$, where R is a uniform distribution independent of W and U_l is the uniform distribution on l -bit strings.*

Definition 2.7 *A family of efficient hash functions $\mathcal{F} = \{h_i : \{0, 1\}^n \rightarrow \{0, 1\}^l\}$ is XOR-universal if $\forall x, y, a, Pr_i[h_i(x) \oplus h_i(y) = a] \leq 2^{-l}$*

From the leftover hash-lemma [10] we can conclude that if $l \leq m' - 2\log(\frac{1}{\epsilon}) + 1$ then a XOR-universal hash function family \mathcal{F} which takes n bits to l bits is a (n, m', l, ϵ) strong extractor.

2.2 Sketches and fuzzy extractors

A *sketch* is a method of securely storing a value w such that it can be recovered by a user as long as the user knows w' which is “close” to w [10].

Definition 2.8 An (m, m', d) sketch is defined by two algorithms Gen and Rec which have the following properties:

1. For all random variables W where $H_\infty(W) \geq m$ we have $H_\infty(W|\text{Gen}(W)) \geq m'$.
2. For all $w, w' \in \mathcal{W}$ such that $\|w - w'\| \leq d$, $\text{Rec}(w', \text{Gen}(w)) = w$.

From the definition alone, a secure sketch may not be sufficient when the adversary knows many sketches $\text{Gen}(w; r)$. However, a *reusable sketch* introduced by Boyen [3], hides the secret w even when given many sketches. Since w is a fuzzy secret, each sketch is on a close variant of w . In order to capture the worst case of these variations, we allow the adversary to receive sketches of $\delta(w)$ for a perturbation δ of the adversary’s choice out of a set of perturbations Δ , a set of efficiently computable functions from \mathcal{M} to \mathcal{M} that (1) contains the identity function and (2) is closed under composition. We will use Δ as a parameter for our security definitions; it is known that reusability for all Δ is unobtainable [3].

Definition 2.9 A (m, m', d, Δ, ν) reusable sketch is an (m, m', d) sketch such that for all state-preserving adversaries \mathbf{A} and random variables W such that $H_\infty(W) \geq m$,

$$\Pr[H_\infty(W|P_0, \dots, P_q) \geq m'] \geq 1 - \nu$$

where $w \leftarrow W$, $P_0 = \text{Gen}(w)$, and for each i , $P_i = \text{Gen}(\delta_i(w))$ where $\delta_i \leftarrow \mathbf{A}(P_{i-1})$, and where q is the number of perturbations \mathbf{A} chooses to specify before halting.

Definition 2.10 Let $\mathcal{C}, \mathbf{C}, \mathbf{C}^{-1}, \mathbf{D}$ be an $[n, k, d]$ linear code. Then $\text{Gen}_{\mathbf{C}}(w; r)$ is defined to be $w \oplus \mathbf{C}(r)$, and $\text{Rec}_{\mathbf{C}}(w', P)$ is defined to be $\mathbf{C}(\mathbf{D}(w' \oplus P)) \oplus P$.

This sketch was shown to be an $(m, m - (n - k), d)$ sketch by Dodis et al. [10] and was shown to be reusable for $\Delta = \{\delta_y : x \mapsto x \oplus y\}$ and $\nu = 0$ by Boyen [3]. We also note it is equivalent to the deterministic linear sketch $\mathbf{H}w$, the *syndrome sketch* of Dodis et al. [9]. We note that in general, a deterministic linear sketch is trivially shown to be reusable as one cannot get multiple sketches of the same secret and all sketches of perturbed values are computable given a non-perturbed sketch due to linearity.

Neither the definition of a sketch nor the definition of a reusable sketch preclude the adversary from modifying sketches. A sketch is said to be *robust* if no adversary can produce a (new) valid sketch after observing *one* valid one.

Definition 2.11 An (m, m', d, ν) robust sketch is an (m, m', d) sketch such that the maximum advantage, $\text{ADV-ROBUST}(\mathbf{A}, \text{Gen}, \text{Rec})$ over all adversaries and (d, m) -pairs is $\leq \nu$, where the advantage is defined as the probability that \mathbf{A} succeeds in the following game:

Setup: $w \leftarrow W$, $w' \leftarrow W'$ where W and W' is a (d, m) -pair and \mathbf{A} receives $P = \text{Gen}(w)$.

Test: $\mathbf{A}(P)$ outputs $P^* \neq P$ and wins if $\text{Rec}(w', P^*) \neq \perp$.

It is important to note that the definitions of robust sketches and reusable sketches provide security assurances against unbounded adversaries.

Using fuzzy sketches, we can create a *fuzzy extractor*, a method of extracting a random-looking string from a random variable W which can be re-constructed from a sketch [10].

Definition 2.12 A (m, l, d, ϵ) fuzzy extractor is given by two algorithms, Fsk and Rep with the following properties:

1. Fsk is a probabilistic algorithm that on input $w \leftarrow W$ where $H_\infty(W) \geq m$ produces (R, P) such that $SD(\langle R, P \rangle, \langle U_l, P \rangle) \leq \epsilon$.
2. $\text{Rep}(w', P) = w''$ is the reproduction procedure with the property that, $\forall w, w' : \|w - w'\| \leq d$ and $R, P \leftarrow \text{Fsk}(w)$, we have $\text{Rep}(w', P) \rightarrow R$.

A fuzzy extractor is very similar to a sketch: Fsk produces P , which can be regarded as the sketch and Rep recovers a value derived from w using P and a value “close” to w .

Definition 2.13 A $(m, l, d, \Delta, \epsilon, \nu)$ reusable fuzzy extractor is a (m, l, d, ϵ) fuzzy extractor such that for all state-preserving adversaries A and for all random variables W such that $H_\infty(W) \geq m$, for all i ,

$$\Pr[SD(\langle R_i, P_0, \dots, P_q \rangle, \langle U_l, P_0, \dots, P_q \rangle) \leq \epsilon] \geq 1 - \nu$$

where $w \leftarrow W$, $P_0 = \text{Gen}(w)$, and for each i , $P_i = \text{Gen}(\delta_i(w))$ where $\delta_i \leftarrow A(P_1, \dots, P_{i-1})$, and where q is the number of perturbations A chooses to specify before halting.

In other words, a fuzzy extractor is *reusable* if the statistical difference ϵ applies even in any attack scenario where multiple related sketches are revealed.

Definition 2.14 A (m, l, d, ϵ, ν) pre-application robust fuzzy extractor is a (m, l, d, ϵ) fuzzy extractor such that the maximum advantage $\text{ADV-ROBUST}_{fe-pre}(A, \text{Fsk}, \text{Rep})$ over all A and (d, m) -pairs W, W' is $\leq \nu$, where $\text{ADV-ROBUST}(A, \text{Fsk}, \text{Rep})$ is defined as the probability that the adversary succeeds in the following game:

Setup: $w \leftarrow W$ and $w' \leftarrow W'$. A receives P where $(P, R) = \text{Fsk}(w)$.

Test: $A(P)$ outputs P^* and succeeds if $\text{Rep}(w', P^*) \neq \perp$ and $P^* \neq P$.

Similarly, an extractor is *post-application robust* if in the setup phase A receives R and P .

2.3 Strongly robust sketches and extractors

In this section we introduce the idea of *strongly robust* sketches and extractors. Previously, the definition of robustness allowed an adversary to see one and only one valid sketch before trying to create a new sketch of a secret w . Strongly robust sketches extend this notion and require that an adversary cannot create a new valid sketch, even after seeing many valid sketches of perturbations of w .

Definition 2.15 Define A 's success probability in the following game with a sketch (Gen, Rec) as $\text{ADV-ROBUST}'(A, \text{Gen}, \text{Rec}, \Delta)$.

Setup: Two values are sampled from a (d, m) pair, $w \leftarrow W, w' \leftarrow W'$.

Queries: For $i = 1 \dots q$, A selects a $\delta_i \in \Delta$, and receives $\text{Gen}(\delta_i(w)) = P_i$.

Test: $A(P_1, P_2, \dots, P_q)$ outputs P^*, δ^* where $\forall i P^* \neq P_i$ and wins if $\text{Rec}(\delta^*(w'), P^*) \neq \perp$.

With the previous definitions in mind, we formally define the notion of both a *statistically strongly robust sketch* as well as a *strongly robust sketch*. The difference is that the former is strongly robust against an unbounded adversary, while the latter is only computationally strongly robust.

Definition 2.16 An (m, m', d, Δ, ν) statistically strongly robust sketch is an (m, m', d) sketch where for all A and all (d, m) pairs, $\text{ADV-ROBUST}'(A, \text{Gen}, \text{Rec}, \Delta)$ is $\leq \nu$.

Definition 2.17 An (m, m', d, Δ, ν) strongly robust sketch is an (m, m', d) sketch where for all probabilistic, polynomial-time A and for all (d, m) pairs, $\text{ADV-ROBUST}'(A, \text{Gen}, \text{Rec}, \Delta)$ is $\leq \nu$.

We can extend the notion of a strongly robust sketch to a strongly robust extractor.

Definition 2.18 An $(m, l, d, \Delta, \epsilon, \nu)$ strongly pre-application robust fuzzy extractor is an $(m, l, d, \Delta, \epsilon, \nu)$ reusable fuzzy extractor such that the maximum advantage $\text{ADV-ROBUST}'_{fe-pre}(A, \text{Fsk}, \text{Rep}, \Delta) \leq \nu$. We define $\text{ADV-ROBUST}'_{fe-pre}(A, \text{Fsk}, \text{Rep}, \Delta)$ as the maximum probability over all (d, m) -pairs of A succeeding in the following game:

Setup: $w \leftarrow W$, and $w' \leftarrow W'$ for a (d, m) pair W, W' .

Queries: For $i = 1, \dots, q$, $A(P)$ makes a query $\delta_i \in \Delta$ and receives P_i where $(P_i, R_i) = \text{Fsk}(\delta_i(w))$.

Test: A outputs P^* and succeeds if $\text{Rep}(w', P^*) \neq \perp$ and $P^* \neq P_i$ for any i .

Definition 2.19 An $(m, l, d, \Delta, \epsilon, \nu)$ strongly post-application robust fuzzy extractor is an $(m, l, d, \Delta, \epsilon)$ reusable fuzzy extractor such that the maximum advantage $\text{ADV-ROBUST}'_{fe-post}(A, \text{Fsk}, \text{Rep}, \Delta) \leq \nu$, where we define $\text{ADV-ROBUST}'_{fe-post}(A, \text{Fsk}, \text{Rep}, \Delta)$ as the maximum probability over all (d, m) -pairs of A succeeding in the following game:

Setup and Test are as in Definition 2.18.

Queries: For $i = 1, \dots, q$, $A(P)$ makes a query $\delta_i \in \Delta$ and receives (P_i, R_i) where $(P_i, R_i) = \text{Fsk}(\delta_i(w))$.

3 Impossibility results

In this section we prove several impossibility results. Specifically, we show that it is impossible to construct a keyless or logarithmic-key, statistically strongly robust sketch under “reasonable” Δ . Our impossibility results easily extend to show that it is impossible to construct keyless or logarithmic-key strongly (pre-application) robust fuzzy extractors for such Δ .

Specifically we consider a set of perturbations Δ to be reasonable if: (1) There is a $\Lambda \subset \Delta$ such that Λ is a group of isometric permutations, that is, permutations such that $\|w - w'\| = \|\delta(w) - \delta(w')\|$, and (2) there is a $\delta^1 \in \Lambda$ such that for all x , $\|x - \delta^1(x)\| = 1$. We feel these assumptions represent any reasonable choice for Δ . Boyen notes certain degenerate types of perturbations for which reusable sketches are impossible, and restricts his general observations about reusable sketches to Δ that contain a group of isometric permutations. As for δ^1 , recall that the adversary’s ability to specify $\delta \in \Delta$ to apply to w is meant to be a worst-case emulation of variance in w ; as such, it would be unreasonable to expect that a low-difference perturbation such as δ^1 cannot be specified.

We first give a review of results originally developed by Boyen [3]. Let $\text{Gen}^*(w)$ be the set $\{P : \exists r : P = \text{Gen}(w; r)\}$. Let $\mathcal{E} \subset \mathcal{W}$ be any subset, and let $\text{Gen}^*(\mathcal{E})$ be the union $\cup_{w \in \mathcal{E}} \text{Gen}^*(w)$. For a sketch P , let $\text{Gen}^{-1}(P) = \{w : \exists r : \text{Gen}(w; r) = P\}$. Similarly, if \mathcal{S} is a set of sketches, let $\text{Gen}^{-1}(\mathcal{S}) = \{w : \exists r \text{ Gen}(w; r) \in \mathcal{S}\}$.

Lemma 3.1 (Boyen [3]) *Let Gen and Rec be an (m, m', d, Δ) reusable sketch where $\Lambda \subset \Delta$ is a group of isometric permutations. Then:*

1. *The reusable sketch Gen and Rec divide \mathcal{M} into at most $2^{m-m'}$ equivalence classes, \mathcal{E}_i where $w, w' \in \mathcal{E}_i$ iff $\text{Gen}^*(w) = \text{Gen}^*(w')$.*
2. *$\forall \delta \in \Lambda, \forall i, \delta(\mathcal{E}_i) = \mathcal{E}_j$ for some j .*
3. *These classes \mathcal{E}_i are determined by the sketch protocol alone.*
4. *For all $i, j, |\mathcal{E}_i| = |\mathcal{E}_j|$. (As such, let $|\mathcal{E}|$ be the size of any class \mathcal{E}).*

To prove that it is impossible to construct a statistically strong robust sketch, we show that any reusable sketch is not statistically strongly robust. The conclusion follows as statistically strongly robust sketches must be reusable else an adversary can view multiple sketches, recover w and make a valid sketch.

We first give the following inefficient attack, which suffices to prove our main impossibility result in the keyless case:

Theorem 3.2 *Let Gen, Rec be a reusable sketch for Δ which contains a group of isometric permutations, and also includes a δ^1 such that $\forall x, \|x - \delta(x)\| = 1$. Then Gen, Rec is not statistically strongly robust for $d > 1$.²*

Proof. Let W, W' be a $(d-1, m)$ -pair, which is always a (d, m) -pair.

The attack is relatively simple. We simply obtain, through queries, the entire class $\text{Gen}^*(w)$. (We make queries until, with high probability, every random tape for Gen will have been used at least once.) $\text{Gen}^*(w)$ uniquely determines \mathcal{E}_i , the equivalence class containing w . Find $\delta^1(\mathcal{E}_i)$. By Lemma 3.1, there is some $\mathcal{E}_j = \delta^1(\mathcal{E}_i)$. Select $w'' \in \mathcal{E}_j$ and find a $P^* \in \text{Gen}^*(w'')$ such that $P^* \notin \text{Gen}^*(\mathcal{E}_i)$, and let δ^* be the identity. We know we can find such a P^* because $\|w - \delta(w)\| = 1 < d$, and the minimum distance between values in \mathcal{E}_i is $\geq d$ (this follows from the fact that \mathcal{E}_i can be thought of as a code, by Lemma 3.1). We know that $P^* \neq P_i$ for any previous i , since all previous sketches were in $\text{Gen}^*(\mathcal{E}_i)$.

As such, $\text{Rec}(w', P^*)$ will output $\delta(w)$ as long as $\|w' - \delta(w)\| \leq d$ as $\text{Rec}(w, P^*)$ must correct d errors. Since $\|w' - w\| \leq d-1$ by our choice of W' , by the triangle inequality, $\|w' - \delta(w)\| \leq d$. \square

This attack works in that it is easy (for an unbounded adversary) to determine the “correct” equivalence class for a given secret $w \in \mathcal{W}$ given all the sketches of w . While this attack is adequate to prove impossibility against an unbounded adversary, it is inefficient for any sketch that is not (nearly) deterministic. In Appendix C we discuss attacks on specific previous constructions and also describe a more efficient general attack. We also note that this attack demonstrates that if an adversary can uniquely determine the correct equivalence class for a given secret w , we need make no additional assumptions on Δ beyond those made by Boyen in [3].

Note that this attack pertains to sketches that do not use a secret key. If a sketch protocol uses a secret key μ then Gen may take that key as input and affects this attack.

²If $d = 0$, our proof may not apply; the construction may be “robust” simply because there exists no different P^* that could be output.

Theorem 3.3 *Let Gen and Rec be a sketch which utilizes a secret key μ , where $|\mu| = O(\log|w|)$. Then if (Gen, Rec) is reusable for Δ that contains a group of isometric permutations including a δ^1 , it is not robust.*

Proof. The attack proceeds similarly to the attack in the previous theorem, with the addition that the adversary guesses at the secret key μ . If A successfully guesses the key, then A can run the previous attack and thus break robustness. The probability of A selecting the correct key is non-negligible in $|w|$ because of the size of μ . \square

We do not give any results concerning a statistically strongly robust sketch which utilizes a secret key larger than logarithmic in size of the secret. This is because sketches are relatively uninteresting when authenticated channels are assumed, as many techniques for key agreement using noisy secrets over an authenticated channel are known.³

3.1 Impossibility results on fuzzy extractors

Our impossibility results imply that no strongly (pre-application) robust fuzzy extractor against unbounded adversaries can be built in the standard way, where part of P can be considered a sketch, and part of Rep consists of recovering w . Validity in such a construction is limited to validity of the sketch, so unless the sketch is strongly robust, the extractor will not be strongly pre-application robust. In addition, we can extend our proof to show the general impossibility of strongly pre-application robust fuzzy extractors against unbounded adversaries, under some additional assumptions about Δ . See Appendix A for details.

4 Strongly robust fuzzy extractors

In this section we give a construction of a strongly post-application robust fuzzy extractor. Our construction is very similar to [7], extending their construction to achieve reusability and strong robustness.

We construct our fuzzy extractor in the common random string (CRS) model. While it would be preferable to construct a fuzzy extractor without resorting to a common string, we note that the only properties we require out of our string is that it is common to all parties involved, that it is random, and that it is resistant to modification by the adversary. Similarly to [7] our common string need only be chosen once when the system is designed, can be hard coded into all software implementing the system or can be chosen by the parties involved in using the sketch, and can be observed (though not modified) by the adversary. We do not believe that this significantly increases the amount of trust required, a view shared by Cramer et al.

Before we define our construction we define the notion of an xor related-key secure MAC.

Definition 4.1 *A family of functions $MAC_k^{rel} : \{0, 1\}^* \rightarrow \{0, 1\}^n$ is an xor-related key secure MAC if the maximum advantage $ADV\text{-}MAC_{RK}(\mathcal{A}, MAC^{rel}, n)$ is negligible in n , where $ADV\text{-}MAC_{RK}(\mathcal{A}, MAC^{rel}, n)$ is defined to be*

$$ADV\text{-}MAC_{RK}(\mathcal{A}, MAC^{rel}, n) = Pr[k \leftarrow \{0, 1\}^n; (x, \sigma, \delta) \leftarrow \mathcal{A}^{\mathcal{O}_k^{rel}} : MAC_{k \oplus \delta}^{rel}(x) = \sigma]$$

Where x was not a query made to \mathcal{O}_k^{rel} and where \mathcal{O}_k^{rel} returns $MAC_{k \oplus \delta}^{rel}(x)$ on input (x, δ) .

³Assuming the existence of exponentially hard cryptography, a polylogarithmic key suffices against any polynomial adversary.

Our first construction makes use of an xor related-key secure MAC. While there has been little theoretical work on constructing provably related-key secure primitives, we do note that some papers (including Cramer et al. [7]) make use of MACs that are related-key secure but *one-time*. The problem of constructing a *multi-use* xor related-key secure MAC is an open problem. There is reason to believe that certain blockciphers such as AES are resistant to xor related-key attacks, which suffices for xor-related key security of a MAC based on such a blockcipher.

We now give our construction. Let $\mathcal{M} = \{0, 1\}^n$ under the Hamming metric. Let \mathbf{Gen} be a deterministic linear sketch. As such, $\mathbf{Gen}(w) = \mathbf{H}w$ for some $n-k \times n$ matrix \mathbf{H} . Note, the syndrome sketch described earlier is such a sketch. Let l be a parameter such that $l \leq \lceil m' - 2 \log \frac{1}{\epsilon} \rceil$. Let \mathbf{S} be a random matrix such that $\frac{\mathbf{H}}{\mathbf{S}}$ is an $n \times n$ matrix of full rank⁴. Let \mathbf{S}_M be the first l_{MAC} rows of \mathbf{S} and let \mathbf{S}_K be the remaining l_{KEY} rows, so $l_{MAC} + l_{KEY} = l$. Let MAC_μ be an xor-related key secure MAC using key μ . We now construct a strongly pre-application robust fuzzy extractor.

Definition 4.2 ($\text{Fsk}(w)$)

1. Let $\mu = \mathbf{S}_M w$.
2. Let $Q = \mathbf{Gen}(w) = \mathbf{H}w$.
3. Let $R = \mathbf{S}_K w$.
4. Let $\tau = \text{MAC}_\mu(Q)$
5. Output $P = (Q, \tau), R$

Definition 4.3 ($\text{Rep}(w', P)$)

1. Run $w'' = \text{Rec}(w', P)$.
2. Set $\mu' = \mathbf{S}_M w''$ and $R = \mathbf{S}_K w''$.
3. Set $\tau' = \text{MAC}_{\mu'}(Q)$.
4. If $\tau = \tau'$ output R else output \perp .

We now prove some theorems bounding the entropy loss on w due to an unbounded adversary seeing multiple sketches. As indicated footnote 4, H and S can be properly created from the common random string with overwhelming probability $1 - \alpha$.

Lemma 4.4 Let \mathcal{E}_i be an equivalence class of the sketch \mathbf{Gen} . Then if $\frac{\mathbf{H}}{\mathbf{S}}$ is of full rank, \mathbf{S} is a bijection from \mathcal{E}_i to $\{0, 1\}^l$.

Proof. We first show that \mathbf{S} is injective. If it is not, then for some $X, X' \in \mathcal{E}_i$ such that $\mathbf{S}X = \mathbf{S}X'$, we show that $\frac{\mathbf{H}}{\mathbf{S}}X = \frac{\mathbf{H}}{\mathbf{S}}X'$. We know that $\mathbf{H}X = \mathbf{H}X'$ by the fact that the equivalence classes \mathcal{E}_i are defined to be all strings which have the same set of sketches. Thus $\frac{\mathbf{H}}{\mathbf{S}}$ is not full rank. Subjectivity comes from the fact that \mathbf{S} is injective and that $\frac{\mathbf{H}}{\mathbf{S}}$ is n by n and of full rank. \square

Corollary 4.5 \mathbf{S}_M can be thought to partition each class \mathcal{E}_i into a partition of “subclasses” \mathcal{T}_i^z where $X \in \mathcal{T}_i^z$ if $X \in \mathcal{E}_i$ and $\mathbf{S}_M X = z$.

Theorem 4.6 Let $\text{Fsk}^*(w) = \{\delta, \text{Fsk}(\delta(w)) \mid \delta \in \Delta\}$. $H_\infty(W \mid \text{Fsk}^*(W), \mathbf{S}'_M w) \geq m' - l_{MAC}$.

⁴We consider our CRS to be of length about $2n^2$, so that with overwhelming probability $1 - \alpha$, we can use the first n linearly independent rows as $\frac{\mathbf{H}}{\mathbf{S}}$.

Proof. We prove this theorem by bounding the min-entropy for $\delta = \delta_0$, then showing that by the linearity of this construction no further min-entropy is lost for $\delta \in \Delta_{\oplus} \neq \delta_0$. With overwhelming probability we may assume $\frac{\mathbf{H}}{\mathbf{S}}$ is of full rank.

By the reusability of Gen we know that Gen divides \mathcal{M} into equivalence classes \mathcal{E}_i such that $\text{Gen}(w_i) = Q_i$ for all $w_i \in \mathcal{E}_i$. By the previous lemma we can say that \mathbf{S} divides each class \mathcal{E}_i into subclasses \mathcal{T}_i^μ where $\forall w, w' \in \mathcal{T}_i^\mu$, $\text{Gen}(w) = \text{Gen}(w')$, $\mathbf{S}_M w = \mathbf{S}_M w'$. Since \mathbf{S}_M is a permutation on each class, for each class \mathcal{E}_i , there are $2^{l_{MAC}}$ classes \mathcal{T}_i^μ per \mathcal{E}_i . As such, by Lemma 3.1 and the previous sentence there are at most $2^{m-m'+l_{MAC}}$ equivalence classes and by setting $m = n$, the maximum entropy, each class is of size $2^{m'-l_{MAC}}$.

We now consider an adversary who makes queries to $\text{Fsk}(\delta(w))$ where $\delta \neq \delta_0$. Since the sketch is deterministic and linear we know that $\text{Gen}(\delta(w))$ can be calculated directly from $\text{Gen}(w)$ and δ . We also know that if $\mu = \mathbf{S}_M w$, then $\mu_{\delta_x} = \mathbf{S}_M \delta_x(w) = \mathbf{S}_M w \oplus \mathbf{S}_M x$. As such, the adversary can pre-calculate the values of Q and τ before making the query and as such the query adds no additional information. \square

Theorem 4.7 (Fsk, Rep) *is an $(m, l_{KEY}, d, \Delta_{\oplus}, \epsilon, \alpha)$ reusable fuzzy extractor.*

Proof. This comes from the fact that $\mathbf{S}w$ is a linear extractor for Δ_{\oplus} and as such $\mathbf{S}_K w$ is statistically close to random, even given τ . We maintain the advantage for $\mathbf{S}_K \delta_x(w)$ as Δ_{\oplus} is a group and as such any adversary capable of calculating $\mathbf{S}_K \delta_x(w)$ can calculate $\mathbf{S}_K w$. \square

Theorem 4.8 (Fsk, Rep) *is a strongly post-application robust fuzzy extractor for $\Delta = \Delta_{\oplus}$ and $\nu = (\max_{A \in PPT}(\text{ADV-MAC}_{\text{RK}}(A, \text{MAC}, l_{MAC})) + 2^{-m'+l_{MAC}} + \alpha)$ against all polynomial-time adversaries.*

Proof. Before giving the reduction, we note that since we are selecting \mathbf{S} randomly, $\mathbf{S}w$ is a universal hash function. Thus, by the leftover hash lemma we can consider μ to be random, even given \mathbf{S} .

We now transform an adversary A who violates post-application robustness to an A' which defeats the related-key security of the MAC. A' plays the part of the challenger, using his oracle to help him create sketches. A' first selects a (d, m) -pair W, W' and samples them to obtain w, w' . When A requests a sketch $\text{Fsk}(\delta_i(w))$, A' computes $Q_i, R_i = \text{Fsk}(\delta_i(w))$, $\mu_i = \mathbf{S}_M \delta_i$, and asks for $\tau_i = \mathcal{O}^{rel}(Q_i, \mu_i)$. A' then returns $P_i = (Q_i, \tau_i)$ to A . Eventually A returns a sketch $(Q^*, \tau^*), \delta^*$ and A' outputs Q^*, τ^* as its forgery under the key offset $S_c \delta^*$.

The main difficulty in the proof is that when A' makes a sketch P of a query w , and asks for a MAC of P , the MAC will, with high probability not be using the key $\mathbf{S}_M w$ and will rather be using a key K .

To overcome this difficulty we note that for the sketch Q there is a secret w' such that $\mathbf{S}_M w' = K$ and that $Q = \text{Gen}(w')$. This comes from the fact that \mathbf{S}_M divides each equivalence class \mathcal{E}_i into subclasses, and there is one subclass for each value K , (because \mathbf{S}_M is surjective). Moreover, the subclass of the secret w is information theoretically hidden just given the sketches. Thus the sketches produced by A' can be considered valid sketches of an appropriately chosen secret w' , and thus A receives a consistent transcript of values. Therefore, A' forges with the same probability as the chance that A breaks reusable robustness, unless A happens to guess the correct value w , or $\frac{\mathbf{H}}{\mathbf{S}}$ is not of full rank a probability which is bounded by $2^{-m'+l_{MAC}} + \prod_{i=0}^{k-1} (1 - \frac{1}{2^{k-i}})$.

By this bound on $\text{ADV-ROBUST}'_{fe-post}$ and Theorem 4.6, we have that (Fsk, Rep) is strongly post-application robust. \square

4.1 Alternative construction using random oracles

Boyer gives an insider-secure fuzzy extractor in the “ q -limited” random oracle model [3]. In this model, the adversary is computationally unlimited, but is limited by q in the number of random oracle queries it can make, both directly, and through its requests to learn sketches. We give a strongly post-application robust extractor in this same model; it differs from our prior construction in that it makes use of any reusable sketch.

Assume \mathcal{O} is a random oracle giving l -bit outputs, and let $l_{TAG} + l_{KEY} = l$. Let (Gen, Rec) be any (m, m', d, Δ) reusable sketch.

Definition 4.9 ($\text{Fsk}(w)$)

1. Compute $Q = \text{Gen}(w)$.
2. Select a random value r .
3. Compute $X = \mathcal{O}(w, P, r)$. Denote the first l_{TAG} bits as τ , the last l_{KEY} bits as R .
4. Output $P = (Q, \tau), R$.

Definition 4.10 ($\text{Rep}(w', P)$)

1. Compute $w'' = \text{Rec}(w', Q)$.
2. Compute $\tau' || R' = \mathcal{O}(w'', P, r)$, where $|\tau'| = l_{TAG}$.
3. If $\tau = \tau'$, output R' , else output \perp .

Theorem 4.11 (Fsk, Rep) is a $(m, l_{KEY}, d, \Delta, 2^{(l_{KEY}-m'+\log(1-q^2 2^{-m'}))/2}, O(q^2)/2^{m'})$ reusable fuzzy extractor in the q -limited random oracle model.

Proof. Since Gen and Rec are a reusable fuzzy sketch, the min-entropy of w given all Q_i values received by the adversary is m' . Consider each tuple (Q_i, τ_i, r_i, R_i) . We claim that the additional values τ_i, R_i do not substantially reduce the min-entropy of w .

Let S be the set of values w such that there is a Q and an r such that the adversary queries $\mathcal{O}(w, Q, r)$, and let T be the set of δ for which the adversary learned a sketch $\delta(w)$. Note that since each δ queried requires a \mathcal{O} query in Fsk , $|T| + |S| \leq q$. Unless $\delta(w)$ is in S for some $\delta \in T$ (which happens with probability at most $O(q^2)/2^{m'}$), all that is learned from these random oracle queries is that $w \notin S$. This information eliminates at most $|T||S|$ values, each with probability at most $2^{-m'}$ given the known Q . Thus, $H_\infty(W|(Q_i, \tau_i, r_i), \mathcal{O}(S)) \geq m' - \log(1 - |T||S|2^{-m'})$. We denote $m' - \log(1 - q^2 2^{-m'})$ as m'' .

A random oracle represents an optimal randomness extractor [3, 15], and thus for each i , and r , $SD(\langle \mathcal{O}(w, Q, r), r, Q \rangle, \langle U_l, r, Q \rangle) \leq \epsilon$ where $\epsilon = \sqrt{2^{l_{KEY}-m''}} = 2^{(l_{KEY}-m'')/2}$. \square

Corollary 4.12 The min-entropy of w , given $\text{Fsk}(w) = Q_i, \tau_i, r_i, R_i$ is $\geq m''$ with overwhelming probability, given only a polynomial number of sketches.

Theorem 4.13 Fsk and Rep is a $(m, l_{KEY}, d, \Delta, 2^{(l_{KEY}-m'+\log(1-q^2 2^{-m'}))/2}, 2^{-m'+\log(1-q^2 2^{-m'})} + 2^{-l_{TAG}} + O(q^2)/2^{m'})$ strongly post-application robust extractor.

Proof. The probability that the adversary makes a successful forgery is the probability that the adversary can either guess the correct w , or that the sketch Q', r', τ' is such $\mathcal{O}(w, Q', r') = \tau'$. By Corollary 4.12 the min-entropy of w given the tuples Q_i, τ_i, r_i, R_i is m'' . For any tuple Q', τ', r' the probability that for a given w , $\mathcal{O}(w, Q', r') = \tau'$ is $\frac{1}{2^{l_{TAG}}}$. Thus, the forgery probability of the adversary is $2^{-m''} + 2^{-l_{TAG}}$. \square

5 Applications to insider security

Boyen [3] introduced the notion of an “insider secure” fuzzy extractor – a fuzzy extractor which is secure even when the adversary is allowed to see the extracted values for adversarially generated sketches and permutations. We prove that our strongly robust fuzzy extractor is insider secure.

Definition 5.1 (Insider security) *A fuzzy extractor Fsk and Rep is considered to be insider secure for an adversary A if $\text{ADV-INSIDE}(A, \text{Fsk}, \text{Rep}, \Delta)$ is negligible, where $\text{ADV-INSIDE}(A, \text{Fsk}, \text{Rep}, \Delta)$ is the probability of A winning in the following game:*

Setup: *The challenger samples W to obtain w .*

Pre-challenge Queries: *The adversary A presents up to q queries to the challenger where each query is either a public or private query.*

Public Queries: *A selects $\delta_i \in \Delta$ and receives P_i where $\text{Fsk}(\delta_i(w)) = P_i, R_i$.*

Private Queries: *The adversary selects $\delta_i \in \Delta$ and a public sketch P'_i and receives $\text{Rep}(\delta_i(w), P'_i) = R_i$.*

Challenge: *The adversary selects any public sketch P^* that was returned via a public query, under the constraint that for all private queries δ_i, P'_i such that $P'_i = P^*$, δ_i must have the property that for all $w \in \mathcal{M}$, $\|\delta_i(w) - w\| > d$.*

Post-challenge Queries: *The adversary may make further private and public queries, with the stipulation that no private query δ_i, P^* can be made unless $\|\delta_i(w) - w\| > d$ for all w .*

Test: *The adversary succeeds if he outputs R^* such that $\text{Rep}(w, P^*) = R^*$.*

We give a construction of an insider secure fuzzy extractor in Appendix D. As a summary, our construction is the construction of Section 4, using the code offset construction of Gen and Rec . This allows us to give a construction where the adversary may simulate all of its private queries. As such, we need only consider adversaries which make only public queries. In such a case, ADV-INSIDE can be bounded by ϵ , since Fsk, Rep is a reusable fuzzy extractor.

Boyen remarked that it was an open problem to create a fuzzy extractor that is insider-secure without the random oracle model [3]. If xor-related key secure MACs exist in the plain model, then our constructions solve this open problem.

6 Conclusion

In this paper we developed the notion of a strongly robust sketch, a sketch that maintains its robustness property even given multiple sketches. While both reusable sketches and robust sketches exist in the literature with statistical security properties, we have proven that a statistically strongly robust sketch cannot exist for reasonable Δ without using long secret keys.

We then gave a (computationally) strongly robust sketch based on the Hamming distance metric, in the CRS model. We proved our sketch secure under the assumption of a related-key secure MAC. We then showed how a strongly robust sketch can be used to create a strongly robust fuzzy extractor, and we demonstrated that this strongly robust fuzzy extractor was computationally insider secure in the plain model.

References

- [1] Charles Bennett, Giles Brassard, Claude Crpeau, and Ueli M. Maurer. Generalized privacy amplification. In *IEEE Transactions on Information Theory*, volume 41, pages 1915 – 1923, 1995.
- [2] Charles Bennett, Giles Brassard, and Jean Marc Robert. Privacy amplification by public discussion. In *SIAM Journal on Computing*, volume 17, pages 210–229, 1988.
- [3] Xavier Boyen. Reusable cryptographic fuzzy extractors. In *ACM Conference on Computer and Communications Security—CCS 2004*, pages 82–91. New-York: ACM Press, 2004.
- [4] Xavier Boyen, Yevgeniy Dodis, Jonathan Katz, Rafail Ostrovsky, and Adam Smith. Secure remote authentication using biometric data. In Ronald Cramer, editor, *Advances in Cryptology - EUROCRYPT*, volume 3494, pages 147–163, May 2005.
- [5] Ee chien Chang, Vadym Fedyukovych, and Qiming Li. Secure Sketch for Multi-Sets. Cryptology ePrint Archive, Report 2006/090, 2006.
- [6] Ee chien Chang and Qiming Li. Small secure sketch for point-set difference. Cryptology ePrint Archive, Report 2005/145, 2005.
- [7] Ronald Cramer, Yevgeniy Dodis, Serge Fehr, Carles Padro, and Daniel Wichs. Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors. In *Advances in Cryptology - EUROCRYPT*, pages 471–488, April 2008.
- [8] Yevgeniy Dodis, Jonathan Katz, Leonid Reyzin, and Adam Smith. Robust fuzzy extractors and authenticated key agreement from close secrets. In *Advances in Cryptography - CRYPTO*, pages 232–250, 2006.
- [9] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *SIAM Journal on Computing*, volume 38, pages 523–540, 2008.
- [10] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate string keys from biometrics and other noisy data. In Cachin and Camenisch, editors, *Advances in Cryptography - Eurocrypt*, volume 3027 of *Lecture Notes in Computer Science*, pages 523–540, 2004.
- [11] Niklas Frykholm and Ari Juels. Error-tolerant password recovery. In *8th ACM conference on Computer and Communications Security*, pages 1 – 9, 2001.
- [12] Ari Juels and Martin Wattenberg. A fuzzy commitment scheme. In *Sixth ACM Conference on Computing and Communication Security*, pages 28–36, 1999.
- [13] Bhavana Kanukurthi and Leonid Reyzin. An improved robust fuzzy extractor. In *SCN*, pages 156–171, 2008.
- [14] Jean paul Linnartz and Pim Tuyls. New shielding functions to enhance privacy and prevent misuse of biometric templates. In *In AVBPA 2003*, pages 393–402, 2003.

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

A Strongly robust fuzzy extractors

In this appendix, we expand on a remark we made earlier about combining reusability and robustness in fuzzy extractors, namely extending our impossibility results on strongly robust sketches to show that strongly robust extractors are impossible.

It is easy to see how this is the case for all “standard” fuzzy extractor constructions, constructions where Fsk outputs a sketch $\text{Gen}(w)$ for some sketch that is later used to recover w by Rep . As the validity of Fsk is based on the validity of the sketch Gen , we can utilize all the results of Lemma 3.1. However, for general constructions, it may not be the case that we can use the results of Boyen with regards to fuzzy sketches. Most notably, we do not know if $\delta \in \Lambda$ maps pre-image sets to pre-image sets, and we do not know if the preimages induced by Fsk form a code.

We can utilize Claims 11.1, 11.2, 11.3, and 11.4 from Boyen to demonstrate that Fsk must split up \mathcal{M} into equivalence classes, and that $\delta \in \Lambda$ must map classes to classes as the proofs of these claims generalize to reusable fuzzy extractors as the claims only rely on the fact that there must be some min-entropy remaining in w , even after seeing all conceivable sketches of w which we know must be the case, else an unbounded adversary could easily break the strong robustness of the fuzzy extractor.

However, if $\text{Rep}(w, \text{Fsk}(w)) = \text{Rep}(w', \text{Fsk}(w'))$ for $w, w' : \|w - w'\| \leq d$, then it may be the case that the equivalence classes induced by Fsk do not form an error correcting code. As long as we allow that there is a δ^k , $k < n$ such that δ^k maps one class to a different class then we can maintain both of the assumptions necessary for Theorem 3.2 and as such break strong robustness of that extractor.

We also note that as long as the fuzzy extractor has the property that we need not see *every* sketch of a secret w before we can find the equivalence class $\text{Fsk}^{-1}(\text{Fsk}(w))$, then we need make no assumptions about δ at all as our more efficient attack in Appendix C will be able to find a valid sketch of w , without having to see all possible valid sketches.

B Protocols utilizing fuzzy sketches and extractors

In this section, we go through previously designed protocols that have utilized fuzzy extractors in order to demonstrate both the utility and necessity of strongly robust sketches and extractors.

Past protocols which utilize fuzzy extractors can be divided into two types, both types involving a single client and server:

Protocol 1:

1. The server stores a sketch P of the client. The server may or may not store the key R extracted from P as well.
2. The client requests P , and uses w, P to regenerate R .
3. Some protocol utilizing the key R is begun between the client and server.

Protocol 2:

1. The client and server have closely correlated secrets w and w' .
2. The client creates a sketch $\text{Fsk}(w) = P$ and sends that to the server.
3. The client and server regenerate a key R from their close secrets and the sketch P .
4. The client and server begin a protocol utilizing the key R .

Protocols of type 1 can be seen, for instance, in [3]. Protocols of type 2 can be seen in [8]. In protocols of type 2, we note that each time the protocol is being run a new sketch is being generated. If the sketch is completely deterministic *and* the client's w is consistently stored, this is not an issue, however sketches for fuzzy extractors must contain at least one randomized component, namely the random seed for the extractor. In protocols of type 1, it is not immediately clear that strong robustness is necessary, as the server stores one, and only one sketch of the client. If a client interacts with multiple servers however, multiple sketches will be created. In addition, consider a man in the middle adversary. In both types of protocols, such an adversary is capable of sending test sketches to the client multiple times, and interacting with the client running the protocol on the resulting key.

For all these reasons, it is reasonable to consider the issue of strong robustness and reusability in fuzzy sketches and extractors. We reiterate that we do not need to consider the issue of a strongly post application robust extractor due to the fact that we do not consider our extractor to maintain security against an unbounded adversary. Since most primitives hide their key to computationally bounded adversaries we may assume that the adversary attacking the robustness of the sketch protocol does not see the key R .

C Further attacks

Our impossibility results in Section 3 demonstrate that previous constructions cannot be statistically strongly robust. In this section, we extend this result and show that previous constructions are not strongly robust even in a computational sense. We give an attack that works on the constructions of [13, 8].

We first give the construction of a robust fuzzy extractor that is found in [13, 8]. Let $\text{SS}(w)$ be a deterministic, linear sketch. As such, there is a matrix \mathbf{S} such that $\text{SS}(w) = \mathbf{S}w$. Let \mathbf{S}' be a matrix such that $\frac{\mathbf{S}}{|\mathbf{S}'|}$ has full rank. Let $\text{SS}'(w) = \mathbf{S}'w$. For $c = \text{SS}'(w)$, let a be the first half of c , b the second half, where both a and b are viewed as elements of $\mathbb{F}_{2^{n'/2}}$. Set $L = 2\frac{k}{n}$. Let $s = \text{SS}(w)$. Pad s such that $|s| = Ln'/2$ and then split s into L bit strings of size $n'/2$. Define $f_{s,i}(x) = x^{L+3} + x^2(s_{L-1}x^{L-1} + s_{L-2}x^{L-2} + \dots + s_0) + ix$. Fsk is now defined as $\text{Fsk}(w) = (s, i, \sigma)$ where i is randomly selected, σ is the last v bits of $f_{s,i}(a) + b$ and where R is the remaining bits of $f_{s,i}(a) + b$.

We give two attacks. The first attacks the post-application robustness of this scheme, the second attacks the pre-application robustness.

- A makes two public queries where $\delta = 0$, receiving s, i, σ, R and s', i', σ', R' (A receives R and R' due to the fact that it is a post-robustness attack).
- A denotes $X = R||\sigma$ and $X' = R'||\sigma'$.
- A computes $X - X' = (i - i')a$ due to the fact that $\text{SS}(w)$ is deterministic and for both queries $\delta = 0$.

- A finds $a = (X - X')(i - i')^{-1} = (i - i')^{-1}(i - i')a = a$.
- A finds $b = X - f_{s,i}(a)$.

Once A finds both a and b and given that SS' is linear and SS is linear, A can easily compute a new sketch for any $\delta \neq 0$ and any i .

This next attack demonstrates that this protocol is not pre-application robust, so as such the adversary does not receive the extracted key R .

- A makes two public queries where $\delta = 0$, receiving s, i, σ and s', i', σ' .
- A computes $\sigma - \sigma' = f_{s,i}(a) + b]_1^v - (f_{s',i'}(a) + b)]_1^v = (i - i')a]_1^v$.
- A makes a new public query where $\delta = 0$, and receives s'', i'', σ'' .
- A creates a new sketch where $s^* = s'', i^* = (i - i') + i''$ and $\sigma^* = \sigma'' + \sigma - \sigma'$.

Due to the fact that for all these public queries, $\delta = 0$ we have that $s = s' = s'' = s^*$. We have $f_{s,i^*}(a) = f_{s,i''}(a) + (i - i')a = f_{s,i''}(a) + \sigma - \sigma'$ as such $\sigma^* = \sigma'' + \sigma - \sigma'$ and so the s^*, i^*, σ^* is a valid sketch.

Next, we describe a general attack that should be more efficient than the one given in our general impossibility result.

Attack. We note that we can break all sketches into two cases: 1) $\text{Gen}^{-1}(P) = \mathcal{E}^*$ is true for only a single class \mathcal{E}^* and 2) $\text{Gen}^{-1}(P) = \mathcal{E}_{i_1} \cup \dots \cup \mathcal{E}_{i_k}$ for some k equivalence classes $k \geq 2$. In case 1, we simply apply the attack from Theorem 3.2. However, for sketches in case 2, we may find a more efficient attack.

If $|\text{Gen}^*(\mathcal{E}_{i_1}) \cap \dots \cap \text{Gen}^*(\mathcal{E}_{i_k})| \geq 2$ then there exists another sketch P^* such that $P^* \in \text{Gen}^*(\mathcal{E}_{i_1}) \cap \dots \cap \text{Gen}^*(\mathcal{E}_{i_k})$, $P^* \neq P$, $P^* \in \text{Gen}^*(\mathcal{E}^*)$ and as such P^* is a valid sketch of w , and we are done.

Therefore, we assume that this is not the case and $|\text{Gen}^*(\mathcal{E}_{i_1}) \cap \dots \cap \text{Gen}^*(\mathcal{E}_{i_k}) \setminus P| = 0$. A then requests another sketch of w , receiving a new P not equal to any prior P . By Lemma 3.1, if it is true that $\text{Gen}^{-1}(P_2) \cap \text{Gen}^{-1}(P) = \mathcal{E}_{i_1} \cup \dots \cup \mathcal{E}_{i_{k'}}$ where $k' < k$, and $\mathcal{E}_{i_j} \subset \text{Gen}^{-1}(P)$ for all j otherwise we contradict our assumption that $\text{Gen}^*(\mathcal{E}_{i_1}) \cap \dots \cap \text{Gen}^*(\mathcal{E}_{i_k})$ contains only P .

Thus, for every sketch that we see, we are able to eliminate at least one equivalence class from the set of possible classes. Denote the current set of q sketches seen by A as \mathcal{P}_q . A continues by examining the current intersection of the images of the possible equivalence classes, excluding all sketches in \mathcal{P}_q . If this set is non-empty, then a new valid sketch of w was found, and we are finished. Otherwise, A requests another sketch, removing at least one equivalence class from consideration. This process continues until we have found either a valid sketch of w , or have determined the correct equivalence class \mathcal{E}^* and as such we can run the attack from Theorem 3.2.

D Proof of insider security

In this section we state our construction of an insider secure fuzzy extractor. Our construction is similar to the construction in Section 4 with the exception that instead of a deterministic linear sketch, we use the code offset sketch of Definition 2.10. Let Gen and Rec be the codeoffset sketch using a code with parity check matrix \mathbf{H} . Let \mathbf{S}_M and \mathbf{S}_K be defined as in Section 4 with respect to this \mathbf{H} .

Definition D.1 (Fsk)

1. $Q = \text{Gen}(w)$.
2. $\mu = \mathbf{S}_M w$.
3. $\tau = \text{MAC}_\mu(Q)$.
4. $R = \mathbf{S}_K w$.
5. Output $P = (Q, \tau)$ and R .

Definition D.2 (Rep(w', P'))

1. Let $w'' = \text{Rec}(w', Q')$.
2. $\mu' = \mathbf{S}_M w''$
3. $\tau' = \text{MAC}_{\mu'}(Q')$
4. If $\tau' = \tau$ output $\mathbf{S}_K w''$ else output \perp .

Theorem D.3 *Our construction in Section 4, replacing Gen and Rec by the code offset sketch is an insider secure fuzzy extractor.*

Proof. This extractor can be shown to be reusable via the same techniques in Theorem 4.5, Theorem 4.4 and Theorem 4.6. We demonstrate how any adversary playing the game defined for the advantage ADV-INSIDE can simulate its private queries. Once we do that, we can limit ourselves to adversaries which make only public queries. The rest of the proof follows from the idea that if an adversary makes only public queries, the game defined for advantage ADV-INSIDE is the same as the reusability of the fuzzy extractor.

There are two cases. If A makes a private query with a Q never returned from a public query he can simulate the result by outputting \perp . This is because if A can create a new Q, τ pair that will not return \perp , then $\text{ADV-ROBUST}'(A, \text{Fsk}, \text{Rep}, \Delta)$ is non-negligible.

We now deal with the case where A makes a private query using a Q, τ, δ_y tuple returned in a public query. The first time this occurs, A can simulate the output by selecting a random l_{KEY} sized bit string R . For all subsequent private queries of this type, if the associated public query was made with δ_x , then A knows the output of Rep will be $R \oplus \mathbf{S}_K(x \oplus y)$ due to the linearity of the extractor.

We next deal with the case where A makes a private query using a Q, τ, δ_x that was returned in a public query, while also specifying a different $\delta_{x'}$. In this case, then $\text{Rec}(w \oplus x', w \oplus x \oplus \mathbf{C}(r)) = w \oplus x \oplus \mathbf{C}(r) \oplus \mathbf{C}(\mathbf{D}(x' \oplus x \oplus \mathbf{C}(r)))$. If $x \oplus x'$ has weight less than d , then $\mathbf{C}(\mathbf{D}(x \oplus x' \oplus \mathbf{C}(r))) = \mathbf{C}(r)$ and as such Rec recovers $w \oplus x$ and as such we are in the previous case. If $x \oplus x'$ has weight greater than d , then $\mathbf{C}(\mathbf{D}(x' \oplus x \oplus \mathbf{C}(r))) = \mathbf{C}(r')$, a different codeword, if the error correcting program D can run at all. As such Rec outputs $w \oplus x \oplus \mathbf{C}(r \oplus r')$. We know that $\tau = \mathbf{S}_M(w \oplus x)$ and as such we can calculate $\mathbf{S}_M \mathbf{C}(r) = \tau \oplus \mathbf{S}_M Q$. This allows us to calculate $\mathbf{S}_M \mathbf{C}(r') = \mathbf{S}_M(w \oplus x \oplus \mathbf{C}(r \oplus r')) \oplus \mathbf{S}_M Q$. As such we know the linear offset between the old value τ , and the new τ' produced by this query, namely $\mathbf{S}_M(\mathbf{C}(r \oplus r'))$. As such, any adversary who can produce a valid τ' has broken the xor related key security of the MAC.

As this covers all the possible private queries, our proof is complete. \square

We note that the main reason why we needed to use the code offset sketch was that we required that $\text{Rec}(w \oplus x', \text{Gen}(w \oplus x))$ produce a known offset of $w \oplus x$, based only on x and x' . Such a “linearity” property is common to all known reusable sketches, including the code offset sketch, its equivalent sketch the “syndrome” sketch, and the generic reusable sketch made by Boyen all of which were shown to be reusable in [3].