Efficient, Reusable Fuzzy Extractors from LWE[*]

Daniel Apon[2], Chongwon Cho[1], Karim Eldefrawy[1] [**], and Jonathan Katz[2]

[1] Information and Systems Science Laboratory, HRL Laboratories
ccho@hrl.com
[2] University of Maryland
{dapon, jkatz}@cs.umd.edu

Abstract. A fuzzy extractor (FE), proposed for deriving cryptographic keys from biometric data, enables reproducible generation of high-quality randomness from noisy inputs having sufficient min-entropy. FEs rely in their operation on a public "helper string" that is guaranteed not to leak too much information about the original input. Unfortunately, this guarantee may not hold when *multiple* independent helper strings are generated from correlated inputs as would occur if a user registers their biometric data with multiple servers; *reusable* FEs are needed in that case. Although the notion of reusable FEs was introduced in 2004, it has received relatively little attention since then.
We first analyze an FE proposed by Fuller et al. (Asiacrypt 2013) based on the learning-with-errors (LWE) assumption, and show that it is *not* reusable. We then show how to adapt their construction to obtain a weakly reusable FE. We also show a generic technique for turning any weakly reusable FE to a strongly reusable one, in the random-oracle model. Finally, we give a direct construction of a strongly reusable FE based on the LWE assumption, that does not rely on random oracles.

1 Introduction

Consider using biometric data as a source for generating cryptographic keys. For example, assume Alice wants to use her biometric data (e.g., fingerprint) w to generate a cryptographic key that she will then use to encrypt her data before storing it on a public server. In a naive approach, Alice could use w itself as the key to encrypt the data. There are two problems with this approach: first, when Alice re-scans her biometric data at a later point in time, it is likely she will recover a value w' that is close, but not equal, to the initial value w. Alice will be unable to recover her original data with such a noisy key if she uses a

[*] This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

[**] Currently at SRI International. Email: karim.eldefrawy@sri.com.

standard encryption scheme. Second, w is not uniform, and thus it is unclear what security is obtained when using w as a key in standard encryption schemes.

Fuzzy extractors. *Fuzzy extractors (FEs)* provide a solution to the above challenges. A fuzzy extractor, first formally introduced by Dodis, Reyzin, and Smith [8], consists of a pair of algorithms $(\mathsf{Gen}, \mathsf{Rec})$ that work as follows: the generation algorithm Gen takes as input a value (e.g., biometric data) w, and outputs (pub, r), where the first of these is called the "helper string." The recovery algorithm Rec takes as input pub along with a value w', and outputs r if w' is "sufficiently close" to the original value w. The security guarantee, roughly speaking, is that r is uniform—or at least computationally indistinguishable from uniform—for an adversary who is given pub, as long as the original input (i.e., w) comes from a distribution with sufficiently high min-entropy.

Fuzzy extractors can address the scenario described earlier. Alice can run Gen on the initial scan of her biometric data to compute $(\mathsf{pub}, r) \leftarrow \mathsf{Gen}(w)$; she will use r to encrypt her data, and send the resulting ciphertext along with pub to the server. When she wishes to recover her data at some later point in time, she will obtain a fresh scan w' of her biometric data and the server will send to Alice both pub and the ciphertext; Alice will compute $r = \mathsf{Rec}(\mathsf{pub}, w')$ and use r to decrypt the ciphertext. This ensures security, even from the server, since the key used for encryption (i.e., r) is uniform even conditioned on pub.

Reusability. One might hope that an FE would remain "secure" even if used on multiple, related input strings $w_1, \ldots$. Concretely, consider a setting in which a user relies on different scans $w_1, \ldots, w_\ell$ of their biometric data when interacting with ℓ servers, where each server (independently) computes $(\mathsf{pub}_i, r_i) \leftarrow \mathsf{Gen}(w_i)$ as above. (We stress that even though the same underlying biometric feature is used every time, the $\{w_i\}$ represent independent scans of that feature; thus, the $\{w_i\}$ will be close to each other but will not necessarily be identical.) Each of the public values pub_i may become known to an adversary, and the original definition of fuzzy extractors does not provide any guarantees in this case. Boyen [6] was the first to highlight this issue, and he showed (contrived) constructions of FEs that are secure when used once, but that completely leak the underlying values $w_1, \ldots, w_\ell$ if used multiple times. Subsequent work of Simoens et al. [16] and Blanton and Aliasgari [4, 5] showed that even some previously proposed constructions of fuzzy extractors are not reusable.

On the positive side, Boyen defined a notion of reusability for FEs (called *outsider security*) and showed that the code-based construction of Dodis et al. [8] is reusable when a linear code is used. (Several variant definitions of reusability have been proposed; we discuss these definitions further in Section 2.2.) Canetti et al. [7] constructed a fuzzy extractor for the Hamming metric whose primary advantage is that it achieves reusability under very weak assumptions on the different scans $w_1, \ldots, w_\ell$. (In contrast, Boyen assumed that $w_i = w \oplus \delta_i$ for a small shift δ_i known to the adversary.) The scheme of Canetti et al. can also be used for sources of lower entropy rate than prior work, if the distribution of the $\{w_i\}$ satisfies a certain assumption. For completeness, however, we note that the scheme of Canetti et al. also has several disadvantages relative to the reusable scheme analyzed by Boyen: it tolerates a lower error rate, has computational—rather than information-theoretic—security, and relies on "composable digital lockers," which in practice would likely be instantiated with a hash function modeled as a random oracle. Alamélou et al. [2] constructed reusable FEs for the set-difference metric, again based on composable digital lockers.

1.1 Our Contributions

In this work, we propose new, indistinguishability-based definitions of reusability for FEs, which can be viewed as adopting aspects of the definitions of reusability given by Boyen [6] and Canetti et al. [7]. We consider both a *weak* and a *strong* notion of reusability. Informally, our definition of weak reusability says that if a sequence of values $(\mathsf{pub}_1, r_1) \leftarrow \mathsf{Gen}(w_1), \ldots$ are computed using related inputs $w_1, \ldots$, then an adversary cannot distinguish r_1 from a uniform string even given all the $\{\mathsf{pub}_i\}$; our notion of strong reusability says that this continues to hold even if the attacker is given $\{r_i\}_{i>1}$.

We then show that a recent computational fuzzy extractor proposed by Fuller at al. [10] (the *FMR scheme*) based on the learning-with-errors (LWE) assumption is not even weakly reusable.[1] In fact, from the public information pub_1 and pub_2 of two instances of the scheme, an attacker can learn the original inputs w_1 and w_2 in their entirety. Fuller et al. do not claim reusability in their paper, but our result is nevertheless interesting as it gives another natural (i.e., not contrived) example of a fuzzy extractor that is not reusable.

We then prove that the FMR scheme *does* achieve weak reusability if common public parameters are used by all parties running the scheme. (Interestingly, the idea of using common public parameters was proposed by Herder et al. [11] for a related scheme, with a different motivation.) Even with this modification, however, the scheme does not achieve *strong* reusability.

With the goal of "bootstrapping" security of the modified FMR scheme from weak to strong reusability, we propose a generic transformation from any weakly reusable FE to a strongly reusable FE in the random-oracle model.[2] Applying this transformation to the modified FMR scheme gives a strongly reusable fuzzy extractor based on the LWE assumption in the random-oracle model.

Finally, we show a construction of a strongly reusable FE based on the LWE assumption that does not rely on the random-oracle model.

1.2 Paper Organization

In Section 2 we review existing definitions of reusable fuzzy extractors and introduce our own definitions of reusability. We analyze the reusability of the FMR scheme in Section 3, showing that it is not weakly reusable as described, but

[1] Huth et al. [12, Theorem 5] claim that the construction of Fuller et al. is reusable, but their proof is incorrect.

[2] Alamélou et al. [2] show a transformation with a similar goal, but it only applies to FEs for the set-difference metric on sets over exponential-size universes.

can be modified to achieve weak reusability. In Section 4, we introduce a generic transformation (in the random-oracle model) that can be applied to any scheme achieving weak reusability in order to obtain strong reusability. We present an LWE-based, strongly reusable FE (without random oracles) in Section 5.

2 Definitions

We let $H_\infty(\cdot)$ denote min-entropy, and let SD denote statistical distance.

2.1 Fuzzy Extractors

Let $\mathcal{M}$ be a metric space with distance metric d. We begin by reviewing the notion of fuzzy extractors (FEs).

Definition 1 (Fuzzy extractor). *Let $\Pi = (\mathsf{Gen}, \mathsf{Rec})$ be such that Gen takes as input $w \in \mathcal{M}$ and outputs (pub, r) with $r \in \{0,1\}^\ell$, and where Rec takes as input pub and $w' \in \mathcal{M}$ and outputs a string $r' \in \{0,1\}^\ell$ or an error symbol $\perp$. We say that Π is an $(\mathcal{M}, \ell, t, \epsilon)$-*fuzzy extractor for class of distributions *$\mathcal{W}$ if:*

Correctness: *For any $w, w' \in \mathcal{M}$ with $d(w, w') \leq t$, if $\mathsf{Gen}(w)$ outputs pub, r, then $\mathsf{Rec}(\mathsf{pub}, w') = r$.*

Security: *For any adversary $\mathcal{A}$ and distribution $W \in \mathcal{W}$, the probability that $\mathcal{A}$ succeeds in the following experiment is at most $1/2 + \epsilon$:*

1. *w is sampled from W, and then $(\mathsf{pub}, r) \leftarrow \mathsf{Gen}(w)$ is computed. Give pub to $\mathcal{A}$.*
2. *Choose $b \leftarrow \{0,1\}$. If $b = 0$, give r to $\mathcal{A}$; otherwise, choose $u \leftarrow \{0,1\}^\ell$ and give u to $\mathcal{A}$.*
3. *$\mathcal{A}$ outputs b', and* succeeds *if $b = b'$.*

The above definition is information-theoretic. For a *computational* $(\mathcal{M}, \ell, t, \epsilon)$-*fuzzy extractor* we require security to hold only for computationally bounded adversaries.

Other models. In this work we will also consider two other models in which FEs can be defined. In the *random-oracle model* we assume that Gen and Rec (as well as the adversary) have access to a uniform function H chosen at the outset of the experiment. In the *public-parameters model* we assume public parameters generated by a trusted party, and made available to Gen and Rec (and the adversary). All our definitions can easily be adapted to either of these models.

2.2 Reusability of Fuzzy Extractors

Definition 1 provides a basic notion of security for FEs. As discussed in the Introduction, however, it does not ensure security if Gen is computed multiple times on the same (or related) inputs. Security in that setting is called *reusability*. Several definitions of reusability have been proposed in prior works [6, 7]. We begin by reviewing prior definitions, and then suggest our own. In all cases, we describe an information-theoretic version of the definition, but a computational version can be obtained in the natural way.

Let $\Pi = (\mathsf{Gen}, \mathsf{Rec})$ be an $(\mathcal{M}, \ell, t, \epsilon)$-fuzzy extractor for class of distributions $\mathcal{W}$. The original definition suggested by Boyen [6, Def. 6] (adapted to the fuzzy extractors rather than fuzzy sketches[3]) considers a set Δ of permutations on $\mathcal{M}$ such that $\delta(w)$ is "close" to w for any $\delta \in \Delta$ and $w \in \mathcal{M}$. It then requires that the success probability of any attacker $\mathcal{A}$ in the following experiment should be small for any distribution $W \in \mathcal{W}$:

1. w^* is sampled from W.
2. $\mathcal{A}$ may adaptively make queries of the following form:
   - $\mathcal{A}$ outputs a perturbation $\delta \in \Delta$.
   - In response, $\mathsf{Gen}(\delta(w^*))$ is run to obtain (pub, r), and $\mathcal{A}$ is given pub.
3. $\mathcal{A}$ outputs w', and *succeeds* if $w' = w^*$.

Informally, then, the attacker is given the public output pub generated by several independent executions of Gen on a series of inputs related (in an adversarially chosen way) to an original value w^*; the definition then guarantees that the attacker cannot learn w^*.

Canetti et al. [7, Def. 2] consider a stronger definition, which requires that the success probability of any attacker $\mathcal{A}$ should be close to $1/2$ in the following experiment:

1. $\mathcal{A}$ specifies a collection of correlated random variables $(W^*, W_2, \ldots, W_\rho)$, where each $W_i \in \mathcal{W}$.
2. Values $w^*, w_2, \ldots, w_\rho$ are sampled from $(W^*, W_2, \ldots, W_\rho)$.
3. Compute $(\mathsf{pub}^*, r^*) \leftarrow \mathsf{Gen}(w^*)$ as well as $(\mathsf{pub}_i, r_i) \leftarrow \mathsf{Gen}(w_i)$ for $2 \leq i \leq \rho$.
4. Give to $\mathcal{A}$ the values pub^* and $\{(\mathsf{pub}_i, r_i)\}_{i=2}^{\rho}$.
5. Choose $b \leftarrow \{0,1\}$. If $b = 0$, give r^* to $\mathcal{A}$; otherwise, choose $u \leftarrow \{0,1\}^\ell$ and give u to $\mathcal{A}$.
6. $\mathcal{A}$ outputs a bit b', and *succeeds* if $b' = b$.

This definition is stronger than Boyen's definition in several respects. First, it allows the attacker to request Gen to be run on a sequence of inputs that are correlated in an arbitrary way with the original value w^*; in fact, there is not even any requirement that w^* be "close" to w_i in any sense. Second, the definition gives the attacker the extracted strings $\{r_i\}$ and not just the public values $\{\mathsf{pub}_i\}$. Finally, it requires that the attacker cannot distinguish the extracted value r^* from a uniform string, rather than merely requiring that the attacker cannot compute the initial input w^*.

Our definitions. The benefit of the definition of Canetti et al. is that it refers to indistinguishability of the extracted string r^* from a uniform string (rather than inability of learning w^* as in Boyen's definition). However, the definition of Canetti et al. seems too strong since it allows for arbitrarily correlated[4] random

[3] A fuzzy sketch [8] is a precursor to a fuzzy extractor, but we do not rely on this notion directly in our work.

[4] Though whether this is realistic depends on whether errors in the biometric readings are dependent or independent of the underlying biometric.

variables $W^*, W_2, \ldots, W_\rho$, rather than adopting a model of perturbations similar to the one considered by Boyen. We combine aspects of the previous definitions to obtain our definitions of *weak* and *strong* reusability, both of which focus on indistinguishability of the extracted string and which restrict the possible perturbations under consideration. Roughly, the definition of weak reusability gives the adversary access to the helper values only, whereas in strong reusability the adversary is also given the extracted strings.

For the next two definitions, we specialize to the Hamming metric for simplicity; for $x \in \mathbb{Z}_q^m$ we let $d(x) = d(x, \mathbf{0})$ denote the Hamming weight (i.e., number of nonzero coordinates) of x.

Definition 2 (Weak reusability). *Let $\Pi = (\mathsf{Gen}, \mathsf{Rec})$ be an $(\mathcal{M}, \ell, t, \epsilon)$-fuzzy extractor for class of distributions $\mathcal{W}$. We say that Π is ϵ-*weakly reusable *if for any $W \in \mathcal{W}$, any adversary $\mathcal{A}$ succeeds with probability at most $1/2 + \epsilon$ in the following experiment:*

1. *A value w^* is sampled from W, and $\mathsf{Gen}(w^*)$ is run to obtain pub^*, r^*. The value pub^* is given to $\mathcal{A}$.*
2. *$\mathcal{A}$ may adaptively make queries of the following form:*
   (a) $\mathcal{A}$ outputs a shift $\delta \in \mathcal{M}$ with $d(\delta) \leq t$.
   (b) $\mathsf{Gen}(w^* + \delta)$ is run to obtain pub and r, and $\mathcal{A}$ is given pub.
3. *Choose $b \leftarrow \{0, 1\}$. If $b = 0$, give r^* to $\mathcal{A}$; otherwise, choose $u \leftarrow \{0, 1\}^\ell$ and give u to $\mathcal{A}$.*
4. *$\mathcal{A}$ outputs a bit b', and succeeds if $b' = b$.*

We remark that weak reusability implies security in the sense of Definition 1.

Definition 3 (Strong reusability). *Let $\Pi = (\mathsf{Gen}, \mathsf{Rec})$ be an $(\mathcal{M}, \ell, t, \epsilon)$-fuzzy extractor for class of distributions $\mathcal{W}$. We say that Π is ϵ-*strongly reusable *if for any $W \in \mathcal{W}$, any adversary $\mathcal{A}$ succeeds with probability at most $1/2 + \epsilon$ in the following experiment:*

1. *A value w^* is sampled from W, and $\mathsf{Gen}(w^*)$ is run to obtain pub^*, r^*. The value pub^* is given to $\mathcal{A}$.*
2. *$\mathcal{A}$ may adaptively make queries of the following form:*
   (a) $\mathcal{A}$ outputs a shift $\delta \in \mathcal{M}$ with $d(\delta) \leq t$.
   (b) $\mathsf{Gen}(w^* + \delta)$ is run to obtain pub and r, which are both given to $\mathcal{A}$.
3. *Choose $b \leftarrow \{0, 1\}$. If $b = 0$, give r^* to $\mathcal{A}$; otherwise, choose $u \leftarrow \{0, 1\}^\ell$ and give u to $\mathcal{A}$.*
4. *$\mathcal{A}$ outputs a bit b', and succeeds if $b' = b$.*

2.3 The Learning-With-Errors Assumption

The learning-with-errors (LWE) assumption was introduced by Regev [15]. We rely on the decisional version of the assumption:

Definition 4 (Decisional LWE). *For an integer $q \geq 2$, and a distribution χ over $\mathbb{Z}_q$, the learning-with-errors problem $\mathsf{LWE}_{n,m,q,\chi}$ is to distinguish between the following distributions:*

$$\{\mathbf{A}, \mathbf{A}\boldsymbol{s} + \boldsymbol{e}\} \text{ and } \{\mathbf{A}, \boldsymbol{u}\}$$

where $\mathbf{A} \leftarrow \mathbb{Z}_q^{m \times n}$, $\boldsymbol{s} \leftarrow \mathbb{Z}_q^n$, $\boldsymbol{e} \leftarrow \chi^m$, and $\boldsymbol{u} \leftarrow \mathbb{Z}_q^m$.

Typically, the error distribution χ under which LWE is considered is the *discrete Gaussian distribution $\mathcal{D}_{\mathbb{Z}_q,\alpha}$* (where α is related to the width of the distribution), but this is not the only possibility.

Akavia et al. [1] showed that LWE has many bits that are simultaneously hardcore. Formally:

Lemma 1 (cf. [1], Lemma 2). *Assume $\mathsf{LWE}_{n-k,m,q,\chi}$ is hard. Then the following pairs of distributions are computationally indistinguishable:*

$$\{\mathbf{A}, \mathbf{A}\boldsymbol{s} + \boldsymbol{e}, \boldsymbol{s}_{1,\ldots,k}\} \text{ and } \{\mathbf{A}, \mathbf{A}\boldsymbol{s} + \boldsymbol{e}, \boldsymbol{u}\}$$

where $\mathbf{A} \leftarrow \mathbb{Z}_q^{m \times n}$, $\boldsymbol{s} \leftarrow \mathbb{Z}_q^n$, $\boldsymbol{e} \leftarrow \chi^m$, and $\boldsymbol{u} \leftarrow \mathbb{Z}_q^k$.

3 Reusability of the FE of Fuller et al.

In this section, we investigate the reusability of a recent construction of a fuzzy extractor due to Fuller et al. [10]. We begin by reviewing the relevant points of their construction, and then show that their scheme is not even *weakly* reusable. (We stress that they do not claim reusability; nevertheless, this result is interesting insofar as it shows a natural example of a fuzzy extractor that is not reusable.) We then show that by making a small modification to their scheme it is possible to achieve weak reusability; even this modified scheme, however, is not *strongly* reusable.

3.1 Background

We first recall the FMR scheme proposed by Fuller, Meng, and Reyzin [10]. Although not directly relevant to our results, we remark that the security of their scheme (in the non-reusable sense) depends on the distribution W over the source. In particular, they proved security based on the LWE assumption when (1) W is the uniform distribution over $\mathbb{Z}_q^n$, as well as when (2) W is a *symbol-fixing* source [13]. Their construction follows the "code-offset" paradigm due to Dodis et al. [8, Section 5], instantiated with a random linear code.

The FMR scheme relies on a subroutine Decode_t that decodes at most $t = O(\log(n))$ errors in a random linear code. Namely, when given a random matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ defining a linear code (with $m \geq 3n$), and a vector $\boldsymbol{b} \in \mathbb{Z}_q^m$ that is guaranteed to be within distance t of a codeword, algorithm Decode_t outputs a vector $\boldsymbol{s} \in \mathbb{Z}_q^n$ such that the codeword $\mathbf{A}\boldsymbol{s}$ is within distance t of $\boldsymbol{b}$. A particular instantiation of this algorithm, that essentially performs a brute-force search for $\boldsymbol{s}$ and is analyzed by Fuller et al., is as follows:

```
$\mathsf{Decode}_t(\mathbf{A}, \boldsymbol{b})$:

1. Select $2n$ distinct indices $i_1, \ldots, i_{2n} \leftarrow \{1, \ldots, m\}$.
2. Restrict $\mathbf{A}, \boldsymbol{b}$ to rows $i_1, \ldots, i_{2n}$; denote these by $\mathbf{A}_{i_1,\ldots,i_{2n}}, \boldsymbol{b}_{i_1,\ldots,i_{2n}}$.
3. Find $n$ linearly independent rows of $\mathbf{A}_{i_1,\ldots,i_{2n}}$. (If no such rows exist, output
   $\perp$ and halt.) Further restrict $\mathbf{A}_{i_1,\ldots,i_{2n}}, \boldsymbol{b}_{i_1,\ldots,i_{2n}}$ to those $n$ rows; denote the
   results by $\mathbf{A}', \boldsymbol{b}'$.
4. Compute $\boldsymbol{s}' = (\mathbf{A}')^{-1}\boldsymbol{b}'$.
5. If $\boldsymbol{b} - \mathbf{A}\boldsymbol{s}'$ has at most $t$ nonzero coordinates, output $\boldsymbol{s}'$. Otherwise, return
   to step 1.
```

Intuitively, decoding works because with good probability step 1 chooses rows where there are no errors; when this occurs, $\boldsymbol{s}' = (\mathbf{A}')^{-1}\boldsymbol{b}'$ is a solution to the linear system and is in fact a unique solution to the original problem. For an analysis of the success probability and (expected) time complexity of Decode_t, we refer readers to the work of Fuller et al. [10].

Assume $w \in \mathbb{Z}_q^m$. The FMR fuzzy extractor is defined as:

```
$\mathsf{Gen}(w)$:

1. Sample uniform $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ and $\boldsymbol{s} \in \mathbb{Z}_q^n$.
2. Let $\mathsf{pub} = (\mathbf{A}, \mathbf{A}\boldsymbol{s} + w)$.
3. Let $r$ be the first $n/2$ coordinates of $\boldsymbol{s}$.
4. Output $(\mathsf{pub}, r)$.
```

```
$\mathsf{Rec}(\mathsf{pub}, w')$:

1. Parse $\mathsf{pub}$ as $(\mathbf{A}, \boldsymbol{c})$; let $\boldsymbol{b} = \boldsymbol{c} - w'$.
2. Compute $\boldsymbol{s}' = \mathsf{Decode}_t(\mathbf{A}, \boldsymbol{b})$.
3. Output $r'$, the first $n/2$ coordinates of $\boldsymbol{s}'$.
```

Note that when w and w' differ in at most t coordinates, $r' = r$ as desired.

3.2 Reusability Analysis

Here we observe that the FMR scheme is not even weakly reusable. In fact, our attack shows that even if the scheme is used only twice, on biometric data w_1, w_2 that are within distance t (but with nothing else about their relation known to the attacker), an attacker can recover w_1 and w_2 in their entirety given the public helper strings $\mathsf{pub}_1, \mathsf{pub}_2$.

In more detail, fix some w_1 and let $w_2 = w_1 + \delta$ where δ has at most t nonzero coordinates. (Both w_1 and δ are unknown to the attacker.) An attacker who observes the public information generated from w_1, w_2 obtains

$$\mathsf{pub}_1 = (\mathbf{A}_1, \mathbf{A}_1\boldsymbol{s}_1 + w_1) \quad \text{and} \quad \mathsf{pub}_2 = (\mathbf{A}_2, \mathbf{A}_2\boldsymbol{s}_2 + w_2).$$

The attacker can then set up the following system of linear equations:

$$\begin{aligned}\mathsf{pub}_1 - \mathsf{pub}_2 &= \mathbf{A}_1 \boldsymbol{s}_1 + w_1 - \mathbf{A}_2 \boldsymbol{s}_2 - w_2\\ &= [\mathbf{A}_1 \mid -\mathbf{A}_2] \cdot \begin{bmatrix}\boldsymbol{s}_1\\ \boldsymbol{s}_2\end{bmatrix} + (w_1 - w_2)\\ &= [\mathbf{A}_1 \mid -\mathbf{A}_2] \cdot \begin{bmatrix}\boldsymbol{s}_1\\ \boldsymbol{s}_2\end{bmatrix} - \delta.\end{aligned}$$

Observe that $[\mathbf{A}_1 \mid -\mathbf{A}_2] \cdot \begin{bmatrix}\boldsymbol{s}_1\\ \boldsymbol{s}_2\end{bmatrix}$ is a linear system with m equations in $2n$ unknowns; moreover, δ has at most t nonzero coordinates. Thus, if $m \geq 6n$ the attacker can use the decoding algorithm described previously to recover $\boldsymbol{s}_1, \boldsymbol{s}_2$ with good probability. The attacker can then compute $w_1 = \mathsf{pub}_1 - \mathbf{A}_1 \boldsymbol{s}_1$ and $w_2 = \mathsf{pub}_2 - \mathbf{A}_2 \boldsymbol{s}_2$. This is a complete break.

In the attack described above we assume $m \geq 6n$, whereas Fuller et al. only require $m \geq 3n$. Is it possible that working in the regime $3n \leq m \leq 6n$ avoids our attack? Unfortunately not. First, one can extend the analysis of Fuller et al. to show that Decode_t works in this regime as well, though with larger expected running time. Moreover, we expect that the attack can be improved given more than two public helper strings.

3.3 Modifying the FMR Scheme

Interestingly, the problem at the heart of the attack shown in the previous section is that the independent enrollments of w_1, w_2 used independently generated matrices $\mathbf{A}_1, \mathbf{A}_2$. Consider instead what happens if a single (randomly generated) matrix $\mathbf{A}$ is used for all enrollments. In this case, proceedings as before gives

$$\begin{aligned}\mathsf{pub}_1 - \mathsf{pub}_2 &= \mathbf{A} \boldsymbol{s}_1 + w_1 - \mathbf{A} \boldsymbol{s}_2 - w_2\\ &= \mathbf{A}(\boldsymbol{s}_1 - \boldsymbol{s}_2) + w_1 - w_2\\ &= \mathbf{A}(\boldsymbol{s}_1 - \boldsymbol{s}_2) - \delta.\end{aligned}$$

An attacker can indeed solve this system of (noisy) linear equations as before; in this case, however, the attacker only learns the difference $\boldsymbol{s}_1 - \boldsymbol{s}_2$ and there is no immediate way for it to obtain w_1, w_2.

In fact, using a common $\mathbf{A}$ can be shown to achieve weak reusability:

Theorem 1. *If the FMR scheme is an $(\mathcal{M}, \ell, t, \epsilon)$-fuzzy extractor, and a common, random $\mathbf{A}$ is used for all executions of* Gen*, then the scheme is also ϵ-weakly reusable.*

Proof. (Sketch) Given $\mathsf{pub} = \mathbf{A}\boldsymbol{s} + w$, with $\boldsymbol{s}, w$ unknown, it is possible to generate a correctly distributed helper string pub_i for the value $w_i = w + \delta_i$ if δ_i is known. Specifically, this can be done by choosing a uniform vector $\boldsymbol{s}_i$ and then setting $\mathsf{pub}_i = \mathbf{A}\boldsymbol{s}_i + \mathsf{pub} + \delta_i$.

Note that since $\mathbf{A}$ is uniform, we in fact only require the existence of a common random string in place of a trusted party who publishes $\mathbf{A}$.

Unfortunately, this modified scheme is not *strongly* reusable. (This is, in fact, also interesting as a natural separation between these two notions of reusability.) Specifically, note that if an attacker computes $\boldsymbol{s}_1 - \boldsymbol{s}_2$ as above, and additionally learns one of the extracted strings r_1 (which, recall, corresponds to the first $n/2$ coordinates of $\boldsymbol{s}_1$), then the attacker can compute r_2.

4 From Weak to Strong Reusability

In this section we show a generic transformation (in the random-oracle model) that can be used to turn any weakly reusable scheme $(\mathsf{Gen}, \mathsf{Rec})$ into a strongly reusable one $(\mathsf{Gen}', \mathsf{Rec}')$. The idea is simply to use $H(r)$ as the extracted string in place of r, where H is a hash function modeled as a random oracle. Intuitively, since $(\mathsf{Gen}, \mathsf{Rec})$ is weakly secure, each of the extracted strings r_i output by Gen is individually uniform; however, there may be correlations between the r_i such that they are not jointly uniform. Applying H to each of the r_i "breaks" these dependencies and enures that the results are jointly uniform.

More formally, let $(\mathsf{Gen}, \mathsf{Rec})$ be an $(\mathcal{M}, \ell, t, \epsilon)$-fuzzy extractor that is weakly reusable, and let $H : \{0,1\}^\ell \to \{0,1\}^\ell$ be a hash function. Let n be a parameter, and construct $(\mathsf{Gen}', \mathsf{Rec}')$ as follows:

$\mathsf{Gen}'(w)$:

1. Compute $(\mathsf{pub}, r) \leftarrow \mathsf{Gen}(w)$.
2. Choose $\mathsf{nonce} \leftarrow \{0,1\}^n$.
3. Output $((\mathsf{pub}, \mathsf{nonce}), H(\mathsf{nonce}, r))$.

$\mathsf{Rec}'((\mathsf{pub}, \mathsf{nonce}), w')$:

1. Compute $r' \leftarrow \mathsf{Rec}(\mathsf{pub}, w')$.
2. Output $H(\mathsf{nonce}, r')$.

Correctness of $(\mathsf{Gen}', \mathsf{Rec}')$ is immediate. We claim that if H is modeled as a random oracle, then $(\mathsf{Gen}', \mathsf{Rec}')$ is strongly reusable (which implies that $(\mathsf{Gen}', \mathsf{Rec}')$ is also a fuzzy extractor in the usual sense).

Theorem 2. *If* $(\mathsf{Gen}, \mathsf{Rec})$ *is an* $(\mathcal{M}, \ell, t, \epsilon)$*-fuzzy extractor, then for any attacker running in time at most* t *it holds that the transformed scheme* $(\mathsf{Gen}', \mathsf{Rec}')$ *is* $O(t \cdot (\epsilon + 2^{-\ell}) + t^2 \cdot 2^{-n})$*-strongly reusable if* H *is modeled as a random oracle.*

Proof. (Sketch) Given an adversary $\mathcal{A}'$ attacking $(\mathsf{Gen}', \mathsf{Rec}')$ in the sense of string reusability, we construct an adversary $\mathcal{A}$ attacking $(\mathsf{Gen}, \mathsf{Rec})$ in the sense of weak reusability. $\mathcal{A}$, given pub^*, chooses a uniform $\mathsf{nonce}^* \in \{0,1\}^n$ and gives $(\mathsf{pub}^*, \mathsf{nonce}^*)$ to $\mathcal{A}'$. When $\mathcal{A}'$ makes a query for a shift δ, adversary $\mathcal{A}$ makes the same query to obtain pub; it then chooses uniform $\mathsf{nonce} \in \{0,1\}^n$ and $r' \in \{0,1\}^\ell$ and gives $((\mathsf{pub}, \mathsf{nonce}), r')$ to $\mathcal{A}'$. Finally, when $\mathcal{A}$ receives a value

$u \in \{0,1\}^\ell$, it gives that value to $\mathcal{A}'$. The H-oracle queries of $\mathcal{A}'$ are simulated by $\mathcal{A}$ in the obvious way.

$\mathcal{A}$ provides a perfect simulation for $\mathcal{A}'$, and u is information-theoretically indistinguishable from uniform, unless (1) a nonce value ever repeats, or (2) $\mathcal{A}'$ ever makes an oracle query of the form $H(\star, r)$, where r is one of the extracted strings (including r^*) output by Gen during the course of the weak reusability experiment involving $\mathcal{A}$. The probability of the first event is easily bounded. Moreover, since each extracted string r is (individually) ϵ-indistinguishable from uniform, the probability of the second event is bounded by $O(t \cdot (\epsilon + 2^{-\ell}))$.

Applying this transformation to the modified FMR scheme from the previous section gives a strongly reusable FE based on the LWE assumption (in the random-oracle model).

5 A Strongly Reusable FE Without Random Oracles

In this section, we construct a strongly reusable FE based on the LWE assumption, without relying on the random-oracle model. The basic idea is to use the scheme of Fuller et al. to encode a random vector $\boldsymbol{s}$ as before; now, however, rather than use some coordinates of $\boldsymbol{s}$ as the extracted string, we instead use $\boldsymbol{s}$ as a key to encrypt an independent random value r. By using a specific symmetric-key encryption scheme based on LWE (that also satisfies certain properties), we obtain a construction based on the LWE assumption.

We first describe our symmetric-key encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$. Let q, m, n be integer parameters. To encrypt a message $\boldsymbol{r} \in \{0,1\}^m$ using a key $\boldsymbol{s} \in \mathbb{Z}_q^n$, choose uniform $\mathbf{B} \leftarrow \mathbb{Z}_q^{m \times n}$ and sample error vector $\boldsymbol{e} \in \mathbb{Z}_q^m$. Finally, compute $\boldsymbol{h} = \mathbf{B}\boldsymbol{s} + \boldsymbol{e} + \frac{q}{2}\boldsymbol{r}$ and output the ciphertext $(\mathbf{B}, \boldsymbol{h})$. To decrypt a ciphertext $(\mathbf{B}, \boldsymbol{h})$ using key $\boldsymbol{s}$, compute $\boldsymbol{h} - \mathbf{B}\boldsymbol{s}$ and then, for each coordinate, output 1 if the coordinate lies in $[\frac{3q}{8}, \frac{5q}{8}]$, and 0 otherwise.

With this in place, we now describe the fuzzy extractor. As previously, we assume a public, random matrix $\mathbf{A}$ available to all participants.

$\mathsf{Gen}(w)$:

1. Choose uniform $\boldsymbol{s} \leftarrow \mathbb{Z}_q^n$.
2. Let $\boldsymbol{c} = \mathbf{A}\boldsymbol{s} + w$.
3. Choose uniform $\boldsymbol{r} \leftarrow \{0,1\}^m$, and compute $(\mathbf{B}, \boldsymbol{h}) \leftarrow \mathsf{Enc}_{\boldsymbol{s}}(\boldsymbol{r})$.
4. Set $\mathsf{pub} = (\boldsymbol{c}, \mathbf{B}, \boldsymbol{h})$.
5. Output (pub, r).

$\mathsf{Rec}(\mathsf{pub}, w')$:

1. Parse pub as $(\boldsymbol{c}, \mathbf{B}, \boldsymbol{h})$; let $\boldsymbol{b} = \boldsymbol{c} - w'$.
2. Compute $\boldsymbol{s}' = \mathsf{Decode}_t(\mathbf{A}, \boldsymbol{b})$.
3. Compute $r' = \mathsf{Dec}_{\boldsymbol{s}'}(\mathbf{B}, \boldsymbol{h})$.
4. Output r'.

Correctness (for the same range of parameters as in the FMR scheme) is immediate. In analyzing security, we assume (1) the distribution on $w \in \mathbb{Z}_q^m$ is such that each coordinate of w is chosen independently according to some distribution χ, and (2) for simplicity, the distribution over the error vector used in our symmetric-key encryption scheme follows the same distribution. (This can in fact be relaxed.) Based on these assumptions, we now sketch a proof of security based on hardness of the $\mathsf{LWE}_{n,(t+2)m,q,\chi}$ problem, where the adversary makes at most t oracle calls in the experiment of Definition 3.

Instantiating Definition 3 with our scheme, we see that we can reframe what we need to prove in the following way. A value w^* is chosen and the attacker $\mathcal{A}$ is given $\mathbf{A}$ and

$$\mathbf{A}\boldsymbol{s} + w^*, \;\; \mathbf{B}, \;\; \mathbf{B}\boldsymbol{s} + \boldsymbol{e} + \frac{q}{2}\boldsymbol{r}^*.$$

Next, for a sequence of shifts δ_i chosen by $\mathcal{A}$, the attacker is given

$$\mathbf{A}\boldsymbol{s}_i + w^* + \delta_i, \;\; \mathbf{B}_i, \;\; \mathbf{B}_i\boldsymbol{s}_i + \boldsymbol{e}_i.$$

(This is not what the attacker is given, but it is easy to see that it is equivalent to what the attacker is given.) The adversary's goal is to distinguish $\boldsymbol{r}^*$ from uniform, and we claim that this is hard.

To see this, consider an algorithm $\mathcal{A}'$ interacting $t+2$ times with an LWE oracle that outputs samples of the form $(\mathbf{B}_i, \mathbf{B}_i\boldsymbol{s} + \boldsymbol{e}_i)$ for a fixed, uniform $\boldsymbol{s}$ chosen at the outset of the experiment. We show how $\mathcal{A}'$ can perfectly simulate the view of $\mathcal{A}$:

1. $\mathcal{A}'$ calls its LWE oracle to obtain $(\mathbf{A}, \boldsymbol{c}^* = \mathbf{A}\boldsymbol{s} + w^*)$, and calls the oracle again to obtain $(\mathbf{B}, \mathbf{B}\boldsymbol{s} + \boldsymbol{e})$. It then chooses a uniform $\boldsymbol{r}^* \in \{0,1\}^m$, sets the public parameters to $\mathbf{A}$, and gives $(\boldsymbol{c}^*, \mathbf{B}, (\mathbf{B}\boldsymbol{s} + \boldsymbol{e}) + \frac{q}{2}\boldsymbol{r}^*)$ to $\mathcal{A}$.
2. When $\mathcal{A}$ submits a shift δ_i, algorithm $\mathcal{A}$ calls its LWE oracle to obtain the pair $(\mathbf{B}_i, \mathbf{B}_i\boldsymbol{s} + \boldsymbol{e}_i)$. It then chooses a uniform $\boldsymbol{s}'_i \in \mathbb{Z}_q^m$ and gives

$$\big((\mathbf{A}\boldsymbol{s} + w^*) + \mathbf{A}\boldsymbol{s}'_i + \delta_i, \;\; \mathbf{B}_i, \;\; (\mathbf{B}_i\boldsymbol{s} + \boldsymbol{e}_i) + \mathbf{B}_i\boldsymbol{s}'_i \big)$$

   to $\mathcal{A}$.

Note that $\mathcal{A}'$ receives $t+2$ outputs from its LWE oracle.

Now consider what happens if the oracle provided to $\mathcal{A}'$ is changed to output pairs of the form $(\mathbf{B}_i, \boldsymbol{u}_i)$, where $\boldsymbol{u}_i$ is uniform. (It is easy to see that distinguishing these two oracles is equivalent to the $\mathsf{LWE}_{n,(t+2)m,q,\chi}$ problem.) In that case, the values given to $\mathcal{A}'$ in the first step take the form $(\boldsymbol{c}^*, \mathbf{B}, \boldsymbol{u} + \frac{q}{2}\boldsymbol{r}^*)$, and we see that $\boldsymbol{r}^*$ is perfectly hidden. Thus, in this modified experiment, $\mathcal{A}$ cannot distinguish $\boldsymbol{r}^*$ from a uniform string. This completes the proof of strong reusability for our construction.

Acknowledgments

We thank Ben Fuller for helpful comments on an earlier version of this work.

References

1. A. Akavia, S. Goldwasser, and V. Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In *6th Theory of Cryptography Conference—TCC 2009*, volume 5444 of *LNCS*, pages 474–495. Springer, 2009.
2. Quentin Alamélou, Paul-Edmond Berthier, Stéphane Cauchie, Benjamin Fuller, and Philippe Gaborit. Reusable fuzzy extractors for the set difference metric and adaptive fuzzy extractors, 2016. Available at http://eprint.iacr.org/2016/1100.
3. Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *J. Mathematical Cryptology*, 9(3):169–203, 2015.
4. Marina Blanton and Mehrdad Aliasgari. On the (non-)reusability of fuzzy sketches and extractors and security in the computational setting. In *Intl. Conf. on Security and Cryptography—SECRYPT 2011*, pages 68–77. SciTePress, 2011. Available at `https://eprint.iacr.org/2012/608`.
5. Marina Blanton and Mehrdad Aliasgari. Analysis of reusability of secure sketches and fuzzy extractors. *IEEE Trans. Information Forensics and Security*, 8(9): 1433–1445, 2013.
6. Xavier Boyen. Reusable cryptographic fuzzy extractors. In *11th ACM Conf. on Computer and Communications Security*, pages 82–91. ACM Press, 2004.
7. Ran Canetti, Benjamin Fuller, Omer Paneth, Leonid Reyzin, and Adam D. Smith. Reusable fuzzy extractors for low-entropy distributions. In *Advances in Cryptology—Eurocrypt 2016, Part I*, volume 9665 of *LNCS*, pages 117–146. Springer, 2016.
8. Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *Advances in Cryptology—Eurocrypt 2004*, volume 3027 of *LNCS*, pages 523–540. Springer, 2004.
9. Nico Döttling and Jörn Müller-Quade. Lossy codes and a new variant of the learning-with-errors problem. In *Advances in Cryptology—Eurocrypt 2013*, volume 7881 of *LNCS*, pages 18–34. Springer, 2013.
10. Benjamin Fuller, Xianrui Meng, and Leonid Reyzin. Computational fuzzy extractors. In *Advances in Cryptology—Asiacrypt 2013, Part I*, volume 8269 of *LNCS*, pages 174–193. Springer, 2013.
11. Charles Herder, Ling Ren, Marten van Dijk, Meng-Day Yu, and Srinivas Devadas. Trapdoor computational fuzzy extractors and stateless cryptographically-secure physical unclonable functions. *IEEE Trans. Dependable and Secure Computing*, 14(1): 65–82, 2017.
12. Christopher Huth, Daniela Becker, Jorge Guajardo, Paul Duplys, and Tim Güneysu. Securing systems with scarce entropy: LWE-based lossless computational fuzzy extractor for the IoT, 2016. Available at http://eprint.iacr.org/2016/982.
13. Jesse Kamp and David Zuckerman. Deterministic extractors for bit-fixing sources and exposure-resilient cryptography. In *44th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 92–101. IEEE, 2003.
14. Ben Lynn, Manoj Prabhakaran, and Amit Sahai. Positive results and techniques for obfuscation. In *Advances in Cryptology—Eurocrypt 2004*, volume 3027 of *LNCS*, pages 20–39. Springer, 2004.
15. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 84–93. ACM Press, 2005.
16. Koen Simoens, Pim Tuyls, and Bart Preneel. Privacy weaknesses in biometric sketches. In *30th IEEE Symp. on Security & Privacy*, pages 188–203. IEEE, 2009.