On the Limitations of Computational Fuzzy Extractors

Kenji Yasunaga* Kosuke Yuzawa†

March 15, 2018

Abstract

We present a negative result of fuzzy extractors with computational security. Specifically, we show that, under a certain computational condition, the existence of a computational fuzzy extractor implies the existence of an information-theoretic fuzzy extractor with slightly weaker parameters. The condition is that the generation procedure of the fuzzy extractor is efficiently invertible by an injective function. Our result implies that to circumvent the limitations of information-theoretic fuzzy extractors, we need to employ computational fuzzy extractors that are not invertible by injective functions.

Keywords: Fuzzy extractor; error-correcting code; computational security.

1 Introduction

Cryptographic primitives generally require uniformly random strings. A *fuzzy extractor* is a primitive proposed by Dodis et al. [4] that can reliably derive uniformly random keys from noisy sources, such as biometric data (fingerprint, iris, facial image, etc.) and long pass-phrases. More formally, a fuzzy extractor is defined to be a pair of procedures $(\mathsf{Gen}, \mathsf{Rep})$. The key generation procedure Gen receives a sample w from a noisy source W with some entropy, and outputs a uniformly random key r and a helper string p. After that, the reproduction procedure Rep can be used to derive the same key r from the helper string p and a sample w' that is close to the original sample w. Notably, this framework does not need secret keys other than w. The derived key r is close to uniform even if the helper string p was given. See [5, 2] for surveys of results related to fuzzy extractors.

To construct fuzzy extractors, Dodis et al. [4] introduced a primitive, called *secure sketch*. On input w, a secure sketch produces a recovery information. It enables the recovery of w from any close value w', but does not reveal much information about w. They show that a combination of a secure sketch and a strong extractor gives a fuzzy extractor.

Fuzzy extractors were defined as *information-theoretic* primitives, and several limitations regarding parameters in fuzzy extractors are also studied in [4]. The *entropy loss* is the difference between the entropy of w and the length of the extracted key r. In the setting of information-theoretic security, the entropy loss is known to be inevitable [8]. This limitation is a major problem for applications using low entropy sources such as biometric data.

Fuller et al. [6] considered the *computational security* of fuzzy extractors to construct *lossless* fuzzy extractors, which circumvent the entropy loss of information-theoretic fuzzy extractors. They

*Institute of Science and Engineering, Kanazawa University. yasunaga@se.kanazawa-u.ac.jp
†Graduate School of Natural Science and Technology, Kanazawa University.

gave both negative and positive results. On one hand, they show that the existence of a computational secure sketch implies the existence of an information-theoretic secure sketch with slightly weaker parameters. This result means that lossless fuzzy extractors cannot be constructed by combining a computational secure sketch and a strong extractor. On the other hand, they present a direct construction of a lossless fuzzy extractor based on the hardness of learning with errors (LWE) problem. The computational security of fuzzy extractors has been studied in subsequent work [3, 1, 7, 9].

In this work, we further study the limitations of computational fuzzy extractors. The negative result of [6] implies that we need to avoid using computational secure sketches to construct lossless fuzzy extractors. However, it remains unclear what properties are necessary for fuzzy extractors to be lossless.

First, we observe that the result of [6] can be applied to computational fuzzy extractors under some condition. The condition is that for the generation procedure Gen, there is an efficient inverter that, on input (r, p), recovers the *same* w that was actually used to generate (r, p) by Gen. It is unclear if the result holds for an inverter without this property. We will discuss this observation in more detail in Section 1.1.

We provide a similar negative result of computational fuzzy extractors under another condition. Specifically, we show that if Gen has an efficient inverter that is *almost injective*, then the existence of a computational fuzzy extractor implies the existence of an information-theoretic fuzzy extractor. This result indicates that a lossless fuzzy extractor must have a property that the generation procedure is not efficiently invertible by injective functions. In the process of proving the result, we fix a flaw in the proof of the result of [6], and obtain a similar lemma with a slightly weaker parameter.

1.1 On the Negative Results of [6]

Fuller et al. noted in [6, footnote 3] that, if the generation procedure Gen is efficiently invertible, their negative results for computational secure sketches can also be applied to computational fuzzy extractors. We observe that this is true if the inverter of Gen satisfies some condition, but it is unclear without it. We describe the observation below in more detail.

Let $(\mathsf{Gen}, \mathsf{Rep})$ be a computational fuzzy extractor. Assume that there is an efficient algorithm InvGen that, given (r, p), outputs w, where (r, p) was generated by $\mathsf{Gen}(w)$. One can construct a computational secure sketch $(\mathsf{SS}, \mathsf{Rec})$ (see Definition 3 for the definition of secure sketches) by defining $\mathsf{SS}(w) = \{(r, p) \leftarrow \mathsf{Gen}(w); \text{Output } p\}$ and $\mathsf{Rec}(w', p) = \{r \leftarrow \mathsf{Rep}(w', p); w \leftarrow \mathsf{InvGen}(r, p); \text{Output } w\}$. Thus, by the negative results of [6], this implies the existence of an information-theoretic fuzzy extractor. However, the above observation can be applied only if $\mathsf{InvGen}(r, p)$ outputs the same w from which (r, p) was actually generated. In general, there could exist different w_1 and w_2 such that the outputs of $\mathsf{Gen}(w_1)$ and $\mathsf{Gen}(w_2)$ are the same. In such a case, one of w_1 and w_2 may not be recovered by InvGen, and thus it may be difficult to use InvGen for constructing secure sketches.

If Gen is injective, then there are no different w_1 and w_2 satisfying $\mathsf{Gen}(w_1) = \mathsf{Gen}(w_2)$, and thus the negative results of [6] can be applied to such computational fuzzy extractors. However, this assumption seems too restrictive. As far as we known, there is no construction of injective fuzzy extractors. Also, there is an intuitive reason for this fact. For a fuzzy extractor $(\mathsf{Gen}, \mathsf{Rep})$, consider two input w_1 and w_2 that are close to each other. If $\mathsf{Gen}(w_1)$ outputs (r, p), then it must be that

$\mathsf{Rep}(w_1, p) = r$ and $\mathsf{Rep}(w_2, p) = r$. Then, it seems natural that the output range of $\mathsf{Gen}(w_2)$ also contains (r, p). If so, the extractor is not injective.

2 Preliminaries

Let X and Y be random variables over some alphabet Z. The *min-entropy* of X is $\mathrm{H}_\infty(X) = -\log(\max_x \Pr[X = x])$. The *average min-entropy* of X given Y is $\tilde{\mathrm{H}}_\infty(X|Y) = -\log(\mathbb{E}_{y \in Z} \max_{x \in Z} \Pr[X = x|Y = y])$. The *statistical distance* between X and Y is $\Delta(X, Y) = \frac{1}{2} \sum_{z \in Z} |\Pr[X = z] - \Pr[Y = z]|$. If $\Delta(X, Y) \leq \epsilon$, we say X and Y are *ϵ-close*. The support of X is $\mathrm{Supp}(X) = \{x \in Z : \Pr[X = x] > 0\}$. We denote by U_ℓ the uniformly distributed random variable on $\{0,1\}^\ell$. For a finite set S, we denote by $t \leftarrow S$ the event that t is chosen uniformly at random from S. For $s \in \mathbb{N}$, the *computational distance* between X and Y is $\Delta^s(X, Y) = \max_{D \in \mathcal{C}_s} |\mathbb{E}[D(X)] - \mathbb{E}[D(Y)]|$, where C_s is the set of randomized circuits of size at most s that output 0 or 1. A metric space is a set $\mathcal{M}$ with a distance function $\mathsf{dis} : \mathcal{M} \times \mathcal{M} \to \mathbb{R}^+ = [0, \infty)$. We always consider finite metric spaces and distance functions with finite images. For the Hamming metric over Z^n, $\mathsf{dis}(x, y)$ is the number of positions in which x and y differ. For a probabilistic experiment E and a predicate P, we denote by $\Pr[E : P]$ the probability that the predicate P is true after the event E occurred. For a probabilistic algorithm A, we denote by $A(x; r)$ the output of A, given x as input and r as random coins.

We give definitions of fuzzy extractor, computational fuzzy extractor, secure sketch, and strong extractor.

Definition 1 (Fuzzy Extractor)**.** *An* $(\mathcal{M}, m, \ell, t, \epsilon)$-fuzzy extractor *with error* δ *is a pair of randomized procedures* $(\mathsf{Gen}, \mathsf{Rep})$ *with the following properties:*

- *The generation procedure* Gen *on input* $w \in \mathcal{M}$ *outputs an extracted string* $r \in \{0,1\}^\ell$ *and a helper string* $p \in \{0,1\}^*$.
- *The reproduction procedure* Rep *takes* $w' \in \mathcal{M}$ *and* $p \in \{0,1\}^*$ *as inputs. The* correctness *property guarantees that for any* $w, w' \in \mathcal{M}$ *with* $\mathsf{dis}(w, w') \leq t$*, if* $(r, p) \leftarrow \mathsf{Gen}(w)$*, then* $\mathsf{Rep}(w', p) = r$ *with probability at least* $1 - \delta$*, where the probability is taken over the coins of* Gen *and* Rep*. If* $\mathsf{dis}(w, w') > t$*, no guarantee is provided about the output of* Rep.
- *The* security *property guarantees that for any distribution* W *on* $\mathcal{M}$ *of min-entropy* m*, if* $(R, P) \leftarrow \mathsf{Gen}(W)$*, then* $\Delta((R, P), (U_\ell, P)) \leq \epsilon$.

Definition 2 (Computational Fuzzy Extractor)**.** *An* $(\mathcal{M}, m, \ell, t, s, \epsilon)$-computational fuzzy extractor *with error* δ *is a pair of randomized procedures* $(\mathsf{Gen}, \mathsf{Rep})$ *that is an* $(\mathcal{M}, m, \ell, t, \epsilon)$*-fuzzy extractor with error* δ *in which the security property is replaced by the following one:*

- *For any distribution* W *on* $\mathcal{M}$ *of min-entropy* m*, if* $(R, P) \leftarrow \mathsf{Gen}(W)$*, then* $\Delta^s((R, P), (U_\ell, P)) \leq \epsilon$.

Definition 3 (Secure Sketch)**.** *An* $(\mathcal{M}, m, \tilde{m}, t)$-secure sketch *with error* δ *is a pair of randomized procedures* $(\mathsf{SS}, \mathsf{Rec})$ *with the following properties:*

- *The sketching procedure* SS *on input* $w \in \mathcal{M}$ *outputs a string* $s \in \{0,1\}^*$.

- *The recovery procedure* Rec *takes* $w' \in \mathcal{M}$ *and* $s \in \{0,1\}^*$ *as inputs. The* correctness *property guarantees that for any* $w, w' \in \mathcal{M}$ *with* $\mathsf{dis}(w, w') \leq t$, $\Pr[\mathsf{Rec}(w', \mathsf{SS}(w)) = w] \geq 1 - \delta$ *where the probability is taken over the coins of* SS *and* Rec. *If* $\mathsf{dis}(w, w') > t$, *no guarantee is provided about the output of* Rec.
- *The* security *property guarantees that for any distribution* W *on* $\mathcal{M}$ *of min-entropy* m, $\tilde{\mathrm{H}}_\infty(W|\mathsf{SS}(W)) \geq \tilde{m}$.

Definition 4. *We say that* $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^\ell$ *is an* (n, m, ℓ, ϵ)-strong extractor *if for any* W *on* $\{0,1\}^n$ *of min-entropy* m, $\Delta((\mathsf{Ext}(W; X), X), (U_\ell, X)) \leq \epsilon$, *where* X *is the uniform distribution on* $\{0,1\}^r$.

3 Limitations of Computational Fuzzy Extractors

In this section, we show that the existence of a computational fuzzy extractor satisfying some computational condition implies the existence of an information-theoretic fuzzy extractor with slightly weaker parameters. The condition is that the generation procedure of a fuzzy extractor is efficiently invertible by an almost-injective function.

We follow a similar approach to Fuller et al. [6], who showed that a computational secure sketch implies an information-theoretic secure sketch. They proved that the existence of a computational secure sketch implies the existence of a code correcting random errors. The result follows by observing that such a code is sufficient to construct an information-theoretic secure sketch [4].

We start from the existence of a computational fuzzy extractor $(\mathsf{Gen}, \mathsf{Rep})$. To show the existence of an error-correcting code, we assume that the generation procedure Gen of the fuzzy extractor is efficiently invertible. The idea for constructing a code is that the inverter of Gen can work as a generator of a codeword from a message. Here, a sample w and an extracted string r from w are considered a codeword and a message, respectively. By fixing the helper string p, we can see that the inverter of Gen is an encoder and the reproduction procedure Rep is a decoder of an error-correcting code. The injectiveness of the inverter of Gen is used to guarantee a high information-rate of the resulting code. The structure used in our approach is different from that in [6]. For a secure sketch $(\mathsf{SS}, \mathsf{Rec})$, they used the fact that by fixing the sketch $ss = \mathsf{SS}(W)$, the procedure of sampling W conditioned on ss is a random sampling of codewords and the recovery procedure Rec can work as a decoder that outputs a corrected codeword, not message.

We give a formal definition of invertibility of the generation procedure.

Definition 5. *Let* $(\mathsf{Gen}, \mathsf{Rep})$ *be a fuzzy extractor for a metric space* $\mathcal{M}$. *We say* Gen *is* (s, η)-invertible *if there exists a deterministic circuit* InvGen *of size at most* s *such that*

$$\Pr\left[W' \leftarrow \mathsf{InvGen}(R', p) : \begin{array}{c} \exists\, r_G \in \{0,1\}^* \ s.t. \\ \mathsf{Gen}(W'; r_G) = (R', p) \end{array}\right] \geq 1 - \eta$$

for any p *that can be generated as* $(r, p) \leftarrow \mathsf{Gen}(w)$ *for* $w \in \mathcal{M}$, *where* $R' = U_\ell$. *In addition, if the inverter* InvGen *has the property such that* $|\{w' : w' \leftarrow \mathsf{InvGen}(U_\ell, p)\}| \geq (1 - \xi)2^\ell$ *for any* p *that can be generated as* $(r, p) \leftarrow \mathsf{Gen}(w)$, *we say* Gen *is* (s, η, ξ)-almost-injectively-invertible.

In the definition, we consider that InvGen succeeds in inverting Gen if it outputs w' from which the input (r', p) can be generated by Gen, and thus w' is not necessarily the same as w from which p was actually generated.

Note that, since the inverter InvGen is confined to being deterministic, InvGen has the property of *output uniqueness*. That is, for any r and p, $\mathsf{InvGen}(r,p)$ does not output two different values $w_1, w_2 \in \mathcal{M}$ such that $(r,p) = \mathsf{Gen}(w_1; r_1) = \mathsf{Gen}(w_2; r_2)$ for some $r_1, r_2 \in \{0,1\}^*$.

We will prove that the existence of a computational fuzzy extractor implies the existence of an error-correcting code. We provide some notions regarding coding theory.

Definition 6. *We say a metric space $(\mathcal{M}, \mathsf{dis})$ is (s,t)-*bounded-error samplable *if there exists a randomized circuit ErrSmp of size s such that for all $0 \le t' \le t$ and $w \in \mathcal{M}$, $\mathsf{ErrSmp}(w,t')$ outputs a random point $w' \in \mathcal{M}$ satisfying $\mathsf{dis}(w,w') = t'$.*

Definition 7. *Let C be a set over a metric space $\mathcal{M}$. We say C is a (t,ϵ)-*maximal-error Shannon code *if there exists an efficient recover procedure Rec such that for all $0 \le t' \le t$ and $c \in C$, $\Pr[\mathsf{Rec}(\mathsf{ErrSmp}(c,t')) \neq c] \le \epsilon$.*

Definition 8. *Let $(\mathcal{M}, \mathsf{dis})$ be a metric space that is (s,t)-bounded-error samplable by a circuit ErrSmp. For a distribution C over $\mathcal{M}$, we say C is a (t,ϵ)-*average-random-error Shannon code *if there exists an efficient recover procedure Rec such that $\Pr[c \leftarrow C, t' \leftarrow \{0,\dots,t\} : \mathsf{Rec}(\mathsf{ErrSmp}(c,t')) \neq c] \le \epsilon$.*

The following fact can be obtained by Markov's inequality.[1]

Lemma 1. *Let C be a (t,ϵ)-average-random-error Shannon code with recovery procedure Rec such that $\mathrm{H}_\infty(C) \ge k$. Then, there exists a set C' with $|C'| \ge 2^{k-1}$ that is $(t, 2\epsilon(t+1))$-maximal-error Shannon code with recovery procedure Rec.*

Proof. Since C is a (t,ϵ)-average-random-error Shannon code, we have that

$$\sum_{c \in \mathrm{Supp}(C)} \Pr[c \leftarrow C] \Pr_{t' \leftarrow \{0,\dots,t\}}[\mathsf{Rec}(\mathsf{ErrSmp}(c,t')) \neq c] \le \epsilon.$$

For $c \in \mathrm{Supp}(C)$, let $\epsilon_c = \Pr_{t' \leftarrow \{0,\dots,t\}}[\mathsf{Rec}(\mathsf{ErrSmp}(c,t')) \neq c]$. By Markov's inequality, it holds that

$$\Pr_{c \leftarrow C}[\epsilon_c \le 2\epsilon] = \Pr_{c \leftarrow C}[\epsilon_c \le 2\mathbb{E}_{c' \leftarrow C}[\epsilon_{c'}]] \ge \frac{1}{2}.$$

Since $\mathrm{H}_\infty(C) \ge k$, there are at least 2^{k-1} codewords $c \in \mathrm{Supp}(C)$ satisfying $\epsilon_c \le 2\epsilon$. Let C' be the set of such codewords. For every $c \in C'$, we have that

$$\sum_{t' \in \{0,\dots,t\}} \Pr[t' \leftarrow \{0,\dots,t\}] \Pr[\mathsf{Rec}(\mathsf{ErrSmp}(c,t')) \neq c] \le 2\epsilon, \tag{1}$$

which implies that $\Pr[\mathsf{Rec}(\mathsf{ErrSmp}(c,t')) \neq c] \le 2\epsilon(t+1)$ for every $t' \in \{0,\dots,t\}$. Otherwise, there exists $t' \in \{0,\dots,t\}$ such that $\Pr[t' \leftarrow \{0,\dots,t\}] \Pr[\mathsf{Rec}(\mathsf{ErrSmp}(c,t')) \neq c] > \frac{1}{t+1} 2\epsilon(t+1) = 2\epsilon$, which contradicts (1). Therefore, C' is a $(t, 2\epsilon(t+1))$-maximal-error Shannon code. □

[1] A similar lemma was given in [6], but the proof has a flaw, which was pointed out by an anonymous reviewer. In their proof, a code was chosen by a probabilistic argument for every $t' \in \{0,\dots,t\}$, but it was not guaranteed that the code is the same for every t'. Instead, we consider a code that corrects random errors for "random" t', which is guaranteed to correct random errors for every t' with a worse decoding error probability.

We prove that if the generation procedure is injectively-invertible, then the existence of a computational fuzzy extractor implies the existence of a maximal-error Shannon code.

Lemma 2. *Let* $(\mathcal{M}, \mathsf{dis})$ *be a metric space that is* (s_{smp}, t)*-bounded-error samplable. Let* $(\mathsf{Gen}, \mathsf{Rep})$ *be an* $(\mathcal{M}, m, \ell, t, s_{\mathsf{sec}}, \epsilon)$*-computational fuzzy extractor with error* 0*. Let* s_{rep} *denote the size of the circuit that computes* Rep*. If* Gen *is* $(s_{\mathrm{inv}}, \eta, \xi)$*-almost-injectively-invertible, and it holds that* $s_{\mathrm{sec}} \geq s_{\mathrm{inv}} + s_{\mathrm{smp}} + s_{\mathrm{rep}}$*, then there exists a value* p *and a set* C *with* $|C| \geq (1-\xi)2^{\ell-1}$ *that is a* $(t, 2(\epsilon+\eta)(t+1))$*-maximal-error Shannon code with recovery procedure* $\mathsf{InvGen}(\mathsf{Rep}(\cdot, p), p)$*.*

Proof. Let W be an arbitrary distribution on $\mathcal{M}$ of min-entropy m. By the security property of the computational fuzzy extractor $(\mathsf{Gen}, \mathsf{Rep})$, we have that $\Delta^{s_{\mathrm{sec}}}((R, P), (U_\ell, P)) \leq \epsilon$ for $(R, P) \leftarrow \mathsf{Gen}(W)$.

Define the following procedure D:

1. On input $r \in \{0,1\}^\ell, p \in \{0,1\}^*$, and $t \in \mathbb{N}$, compute $w \leftarrow \mathsf{InvGen}(r, p)$.
2. $t' \leftarrow \{0, \ldots, t\}$.
3. $w' \leftarrow \mathsf{ErrSmp}(w, t')$.
4. If $\mathsf{Rep}(w', p) \neq r$, output 0. Otherwise, output 1.

The procedure D "efficiently" checks whether Rep can correctly output the string r from the corresponding p and w with random t-bounded errors. We need the efficiency of D since otherwise the "error-correcting" property of Rep may not be taken over from the computational security of $(\mathsf{Gen}, \mathsf{Rep})$. The procedure D can be implemented by a circuit of size $s_{\mathrm{inv}} + s_{\mathrm{smp}} + s_{\mathrm{rep}}$.

By the invertibility of Gen and the correctness property of $(\mathsf{Gen}, \mathsf{Rep})$, we have that $\Pr[D(R, P, t) = 1] \geq 1 - \eta$, where $(R, P) \leftarrow \mathsf{Gen}(W)$. Since $\Delta^{s_{\mathrm{sec}}}((R, P), (U_\ell, P)) \leq \epsilon$, if $s_{\mathrm{sec}} \geq s_{\mathrm{inv}} + s_{\mathrm{smp}} + s_{\mathrm{rep}}$, it holds that

$$\Pr[D(U_\ell, P, t) = 1] \geq 1 - (\epsilon + \eta).$$

By the averaging argument, there exists a value p such that $\Pr[D(U_\ell, p, t) = 1] \geq 1 - (\epsilon + \eta)$. This implies that

$$\Pr\left[\begin{array}{l} w \leftarrow \mathsf{InvGen}(R, p), \\ t' \leftarrow \{0, \ldots, t\}, \\ w' \leftarrow \mathsf{ErrSmp}(w, t') \end{array} : \mathsf{Rep}(w', p) = R \right] \geq 1 - (\epsilon + \eta), \tag{2}$$

where $R = U_\ell$. Thus, the distribution $\mathsf{InvGen}(U_\ell, p)$ is a $(t, \epsilon + \eta)$-average-random-error Shannon code with recovery procedure $\mathsf{InvGen}(\mathsf{Rep}(\cdot, p), p)$. By applying Lemma 1, we can show that there is a set C with $|C| \geq 2^{k-1}$ that is a $(t, 2(\epsilon + \eta)(t+1))$-maximal-error Shannon code for $k \geq \mathrm{H}_\infty(\mathsf{InvGen}(U_\ell, p))$.

It follows from the almost-injective invertibility of Gen that $|\{w' : w' \leftarrow \mathsf{InvGen}(U_\ell, p)| \geq (1-\xi)2^\ell$. Thus, $\mathrm{H}_\infty(\mathsf{InvGen}(U_\ell, p)) \geq \ell - \log(1/(1-\xi))$. Therefore, the statement follows. □

It is known that a secure sketch can be constructed from a Shannon code, which is explicitly presented in [6], and implicitly stated in [4, Section 8.2].

Lemma 3 ([4, 6]). *For an alphabet Z, let C be a (t, δ)-maximal-error Shannon code over Z^n. Then, there exists a $(Z^n, m, m - (n \log |Z| - \log |C|), t)$ secure sketch with error δ for the Hamming metric over Z^n.*

An information-theoretic fuzzy extractor can be constructed from a secure sketch and a strong extractor [4]. In particular, if we use universal hashing as strong extractor, we obtain the following result.

Lemma 4 ([4]). *Let $(\mathsf{SS}, \mathsf{Rec})$ be an $(\mathcal{M}, m, \tilde{m}, t)$-secure sketch with error δ, and Ext an $(n, \tilde{m}, \ell, \epsilon)$-strong extractor given by universal hashing (any $\ell \leq \tilde{m} - 2\log(\frac{1}{\epsilon}) + 2$ can be achieved). Then, the following $(\mathsf{Gen}, \mathsf{Rep})$ is an $(\mathcal{M}, m, \ell, t, \epsilon)$-fuzzy extractor with error δ:*

- $\mathsf{Gen}(w; r, x)$ *: set* $P = (\mathsf{SS}(w; r), x)$, $R = \mathsf{Ext}(w; x)$, *and output* (R, P).
- $\mathsf{Rep}(w', (s, x))$ *: recover* $w = \mathsf{Rec}(w', s)$ *and output* $R = \mathsf{Ext}(w; x)$.

By combining Lemmas 2, 3, and 4, we obtain the following theorem.

Theorem 1. *Let Z be an alphabet. Let $(\mathsf{Gen}, \mathsf{Rep})$ be a $(Z^n, m, \ell, t, s_{\mathsf{sec}}, \epsilon)$-computational fuzzy extractor with error 0. Let s_{rep} denote the size of the circuit that computes Rep. If Gen is $(s_{\mathrm{inv}}, \eta, \xi)$-almost-injectively-invertible, and it holds that $s_{\mathrm{sec}} \geq s_{\mathrm{inv}} + n \log |Z| + s_{\mathrm{rep}}$, then there exists a $(Z^n, m, \ell', t, \epsilon')$ (information-theoretic) fuzzy extractor with error $2(\epsilon + \eta)(t + 1)$ for any $\ell' \leq m + \ell - n \log |Z| - \log(\frac{1}{1-\xi}) - 2\log(\frac{1}{\epsilon'}) + 1$.*

In particular, in the above theorem, if we choose $m = n \log |Z|$, then a $(Z^n, n \log |Z|, \ell, t, s_{\mathsf{sec}}, \epsilon)$-computational fuzzy extractor implies a $\left(Z^n, n \log |Z|, \ell - \log\left(\frac{1}{1-\xi}\right) - 2\log\left(\frac{1}{\epsilon'}\right) + 1, t, \epsilon'\right)$-fuzzy extractor with error $2(\epsilon + \eta)(t + 1)$.

As in the negative result of [6], we do not claim about the efficiency of the resulting fuzzy extractor. In our case, the non-explicit parts are (1) fixing the value p in Lemma 2, and (2) constructing a maximal-error Shannon code from an average-random-error one in Lemma 1.

Acknowledgment

The authors are grateful to Masahiro Mambo for his helpful comments.

This work was supported in part by JSPS/MEXT Grant-in-Aid for Scientific Research Numbers 23500010, 23700010, 24240001, 25106509, 15H00851, 16H01705, and 17H01695.

References

[1] D. Apon, C. Cho, K. Eldefrawy, and J. Katz. Efficient, reusable fuzzy extractors from LWE. In S. Dolev and S. Lodha, editors, *Cyber Security Cryptography and Machine Learning - First International Conference, CSCML 2017, Beer-Sheva, Israel, June 29-30, 2017, Proceedings*, volume 10332 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2017.

[2] X. Boyen. Robust and reusable fuzzy extractor. In P. Tuyls, B. Skoric, and T. Kevenaar, editors, *Security with Noisy Data*, pages 101–112. Springer, 2007.

[3] R. Canetti, B. Fuller, O. Paneth, L. Reyzin, and A. D. Smith. Reusable fuzzy extractors for low-entropy distributions. In M. Fischlin and J. Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 117–146. Springer, 2016.

[4] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139, 2008.

[5] Y. Dodis, L. Reyzin, and A. Smith. Fuzzy extractors. In P. Tuyls, B. Skoric, and T. Kevenaar, editors, *Security with Noisy Data*, pages 79–99. Springer, 2007. An updated version is available at http://www.cs.bu.edu/~reyzin/fuzzysurvey.html.

[6] B. Fuller, X. Meng, and L. Reyzin. Computational fuzzy extractors. In *ASIACRYPT (1)*, pages 174–193, 2013.

[7] C. Herder, L. Ren, M. van Dijk, M. M. Yu, and S. Devadas. Trapdoor computational fuzzy extractors and stateless cryptographically-secure physical unclonable functions. *IEEE Trans. Dependable Sec. Comput.*, 14(1):65–82, 2017.

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

[9] Y. Wen, S. Liu, and S. Han. Reusable fuzzy extractor from the decisional Diffie-Hellman assumption. *Des. Codes Cryptography*, 2018. https://doi.org/10.1007/s10623-018-0459-4.