IEICE TRANS. FUNDAMENTALS, VOL.E106–A, NO.3 MARCH 2023

LETTER *Special Section on Cryptography and Information Security*

On the Limitations of Computational Fuzzy Extractors

Kenji YASUNAGA[†a)], ***Member and*** **Kosuke YUZAWA**[††], ***Nonmember***

SUMMARY We present a negative result of fuzzy extractors with computational security. Specifically, we show that, under a computational condition, a computational fuzzy extractor implies the existence of an information-theoretic fuzzy extractor with slightly weaker parameters. 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 non-lossy functions.

key words: *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. [1] 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 consists of 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 [2], [3] for surveys of results related to fuzzy extractors.

Dodis et al. [1] introduced a primitive, called a *secure sketch*, to construct fuzzy extractors. On input w, a secure sketch produces recovery information. It enables the recovery of w from any close value w', but does not reveal much information about w. They showed that a combination of a secure sketch and a strong extractor gives a fuzzy extractor, where the *strong* extractor means that the seed can be used as a part of the output [1].

Fuzzy extractors were firstly introduced as *information-theoretic* primitives, and several limitations regarding parameters in fuzzy extractors were also studied in [1]. 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, entropy loss is known to be inevitable [4]. This limitation is a major problem for applications using low entropy sources such as biometric data.

Fuller et al. [5] considered the *computational security* of fuzzy extractors to construct *lossless* fuzzy extractors, which circumvent the entropy loss of information-theoretic fuzzy extractors. They gave both negative and positive results. As a negative result, they showed that a computational *secure sketch* implies the existence of an information-theoretic secure sketch with slightly weaker parameters. The result indicates that combining a computational secure sketch and a strong extractor may not give lossless fuzzy extractors. As a positive result, they directly constructed 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 [6]–[12].

In this work, we further study the limitations of computational fuzzy extractors. First, we observe that the negative result of [5] can be applied to computational *fuzzy extractors* under a specific condition. The condition is that the generation procedure Gen is efficiently and *uniquely* invertible in a sense such that, on input (r, p), the inverter recovers the *same* w that was used to generate (r, p) by Gen. See Sect. 1.1 for details.

Next, as a negative result, we show that computational fuzzy extractors imply information-theoretic fuzzy extractors if Gen is efficiently invertible by non-lossy functions. This condition includes the case that the inverter may recover different w' than the original input w. Thus, we extend a negative result of [5] by relaxing the uniqueness requirement of inverters of Gen. In proving the result, we fix a flaw in a proof in [5].

1.1 On the Negative Results of [5]

Fuller et al. noted in [5, 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 the claim 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 sketch) by defining

Manuscript received March 11, 2022.
Manuscript revised June 21, 2022.
Manuscript publicized August 10, 2022.

[†]The author is with the Department of Mathematical and Computing Science, Tokyo Institute of Technology, Tokyo, 152-8552 Japan.
[††]The author was a student at Kanazawa University, Kanazawa-shi, 920-1192 Japan.
a) E-mail: yasunaga@c.titech.ac.jp

DOI: 10.1587/transfun.2022CIL0001

$\mathsf{SS}(w) = \{(r,p) \leftarrow \mathsf{Gen}(w); \text{Output } p\}$ 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 [5], 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. A standard construction of [1] using universal hashing is the case. In such a case, one of w_1 and w_2 may not be recovered by InvGen, and thus it is 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 [5] can be generally applied to such computational fuzzy extractors. However, this assumption seems too restrictive. As far as we know, 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 inputs 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

For $t \in \mathbb{N}$, we write $[0:t] = \{0,1,\ldots,t\}$. 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 $\Delta(X,Y) \le \epsilon$, 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 $a \leftarrow S$ the event that a is chosen uniformly at random from S. For $s \in \mathbb{N}$, the *computational distance* between X and Y is $\Delta^s(X,Y) = \max_{D\in C_s} |\mathbb{E}[D(X)] - \mathbb{E}[D(Y)]|$, 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 experiment 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 $\mathsf{dis}(w,w') \le t$, 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 $\Delta((R,P),(U_\ell,P)) \le \epsilon$.*

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 $\Delta^s((R,P),(U_\ell,P)) \le \epsilon$.*

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 $\mathsf{dis}(w,w') \le t$, $\Pr[\mathsf{Rec}(w',\mathsf{SS}(w)) = w] \ge 1-\delta$, 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, $\tilde{\mathrm{H}}_\infty(W|\mathsf{SS}(W)) \ge \tilde{m}$.*

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, $\Delta((\mathsf{Ext}(W;X),X),(U_\ell,X)) \le \epsilon$, where X is the uniform distribution on $\{0,1\}^r$.*

3. Limitations of Computational Fuzzy Extractors

We show that a computational fuzzy extractor satisfying some condition implies the existence of an information-theoretic fuzzy extractor with slightly weaker parameters.

We follow a similar approach to Fuller et al. [5], who showed that a computational secure sketch implies an information-theoretic secure sketch. They proved that the existence of a computational secure sketch implies an error-correcting code for random errors. The result follows by observing that such a code is sufficient to construct an information-theoretic secure sketch [1].

Our result also needs some invertibility condition on Gen, which does not require the unique invertibility as described in Sect. 1.1. Intuitively, our condition is that Gen is efficiently invertible by non-lossy function. The following is the formal definition.

Definition 5. *Let $(\mathsf{Gen},\mathsf{Rep})$ be a fuzzy extractor for a metric*

space $\mathcal{M}$. *We say* Gen *is* (s,η,ξ)-invertible *if*

1. *there exists a deterministic circuit* InvGen *of size at most* s *such that*

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

   for any p *that can be generated as* $(r,p) \leftarrow \mathsf{Gen}(w)$ *for* $w \in \mathcal{M}$, *where* $R = U_\ell$;
2. *in addition,* InvGen *satisfies that*

$$|\{w' : w' \leftarrow \mathsf{InvGen}(R,p)\}| \geq (1-\xi)2^\ell$$

   for any p *that can be generated as* $(r,p) \leftarrow \mathsf{Gen}(w)$ *for* $w \in \mathcal{M}$, *where* $R = U_\ell$.

In Definition 5, 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.

The first condition of Definition 5 is a natural invertibility condition by considering the roles of r and p. As an attacker (inverter) A of the fuzzy extractor (Gen, Rep), the string r is an extracted random string A may not be accessible, and the helper string p is public information assumed to be unchanged. Thus, the first condition captures the success of attacks by A for the task of inverting Gen.

The second condition excludes the possibility of InvGen to output some "easy-to-answer" value w^*. This non-lossy condition seems necessary to capture the situation that the recovered w' may differ from the original w. The attacker may always output specific w^* for most input (r,p). However, such an attack seems unsuccessful because the output distribution of w^* has low entropy, while input w should have enough entropy to be extracted.

(1) Proof Idea

We start from the existence of a computational fuzzy extractor (Gen, Rep). The idea for constructing an error-correcting code is that an efficient 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 second condition on 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 [5]. For a secure sketch (SS, 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.

(2) Coding Theory

We provide some notions and a technical lemma 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* $0 \leq t' \leq t$ *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 recovery procedure* Rec *such that for all* $0 \leq t' \leq t$ *and* $c \in C$, $\Pr[\mathsf{Rec}(\mathsf{ErrSmp}(c,t')) \neq c] \leq \epsilon$.

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 recovery procedure* Rec *such that* $\Pr[c \leftarrow C, t' \leftarrow [0:t] : \mathsf{Rec}(\mathsf{ErrSmp}(c,t')) \neq c] \leq \epsilon$.

The following is obtained by Markov's inequality[†].

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

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

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

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

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

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

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

(3) Our Negative Result

We prove that if the generation procedure is invertible, then the existence of a computational fuzzy extractor implies the

[†]A similar lemma was given in [5], but the proof has a flaw, which an anonymous reviewer pointed out. In their proof, a code was chosen by a probabilistic argument for every $t' \in [0:t]$. However, it is 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.

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* $(\mathcal{M}, m, \ell, t, s_{\mathrm{sec}}, \epsilon)$*-computational fuzzy extractor with error* 0*. Let* s_{rep} *denote the size of the circuit that computes* Rep*. If* Gen *is* $(s_{\mathrm{inv}}, \eta, \xi)$*-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. $t' \leftarrow [0 : t]$.
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

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

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 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 [5], and implicitly stated in [1, Sect. 8.2].

Lemma 3 ([1], [5])**.** *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 [1]. In particular, if we use universal hashing as strong extractor, we obtain the following result.

Lemma 4 ([1])**.** *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* $(Z^n, m, \ell, t, s_{\mathrm{sec}}, \epsilon)$*-computational fuzzy extractor with error* 0*. Let* s_{rep} *denote the size of the circuit that computes* Rep*. If* Gen *is* $(s_{\mathrm{inv}}, \eta, \xi)$*-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 $(Z^n, n \log |Z|, \ell, t, s_{\mathrm{sec}}, \epsilon)$-computational fuzzy extractor implies a $\left(Z^n, n \log |Z|, \ell - \log(\frac{1}{1-\xi}) - 2\log(\frac{1}{\epsilon'}) + 1, t, \epsilon'\right)$-fuzzy extractor with error $2(\epsilon+\eta)(t+1)$.

As in the negative result of [5], we do not claim 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.

Acknowledgments

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] 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., vol.38, no.1, pp.97–139, 2008.

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

[3] X. Boyen, "Robust and reusable fuzzy extractor," Security with Noisy Data, P. Tuyls, B. Skoric, and T. Kevenaar, ed., pp.101–112, Springer, 2007.
[4] J. Radhakrishnan and A. Ta-Shma, "Bounds for dispersers, extractors, and depth-two superconcentrators," SIAM J. Discrete Math., vol.13, no.1, pp.2–24, 2000.
[5] B. Fuller, X. Meng, and L. Reyzin, "Computational fuzzy extractors," ASIACRYPT (1), pp.174–193, 2013.
[6] R. Canetti, B. Fuller, O. Paneth, L. Reyzin, and A.D. Smith, "Reusable fuzzy extractors for low-entropy distributions," 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, M. Fischlin and J. Coron, eds., Lecture Notes in Computer Science, vol.9665, pp.117–146, Springer, 2016.
[7] D. Apon, C. Cho, K. Eldefrawy, and J. Katz, "Efficient, reusable fuzzy extractors from LWE," Cyber Security Cryptography and Machine Learning - First International Conference, CSCML 2017, Beer-Sheva, Israel, June 29–30, 2017, Proceedings, S. Dolev and S. Lodha, ed., Lecture Notes in Computer Science, vol.10332, pp.1–18, Springer, 2017.
[8] 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 Secure Comput., vol.14, no.1, pp.65–82, 2017.
[9] Y. Wen, S. Liu, and S. Han, "Reusable fuzzy extractor from the decisional Diffie-Hellman assumption," Des. Codes Cryptogr., vol.86, pp.2495–2512, 2018. https://doi.org/10.1007/s10623-018-0459-4
[10] Y. Wen and S. Liu, "Reusable fuzzy extractor from LWE," Information Security and Privacy - 23rd Australasian Conference, ACISP 2018, Wollongong, NSW, Australia, July 11–13, 2018, Proceedings, W. Susilo and G. Yang, eds., Lecture Notes in Computer Science, vol.10946, pp.13–27, Springer, 2018.
[11] Y. Wen and S. Liu, "Robustly reusable fuzzy extractor from standard assumptions," Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, Dec. 2–6, 2018, Proceedings, Part III, T. Peyrin and S.D. Galbraith, eds., Lecture Notes in Computer Science, vol.11274, pp.459–489, Springer, 2018.
[12] Y. Wen, S. Liu, and D. Gu, "Generic constructions of robustly reusable fuzzy extractor," Public-Key Cryptography - PKC 2019 - 22nd IACR International Conference on Practice and Theory of Public-Key Cryptography, Beijing, China, April 14–17, 2019, Proceedings, Part II, D. Lin and K. Sako, eds., Lecture Notes in Computer Science, vol.11443, pp.349–378, Springer, 2019.